AALBORG UNIVERSITET

ESBJERG

Probabilistic transmission model for airborne diseases using agent-based modelling

A programming solution based on movement simulation

Master's thesis – MSc. Risk and Safety Management

Mirco Hennings

January 2022

Aalborg University – Title page

Theme:
Master's thesis

Title:
Probabilistic transmission model for airborne diseases using agent-based modelling

Author: Mirco Hennings

Project group:
RISK4-10

Project period:
Autumn semester 2021

ECTS:
30

Supervisors:
Michael Havbro Faber, José Guadalupe Rangel-Ramirez

Total pages:	91
Pages text:	46
Pages appendix:	35

Abstract:

The current SARS-COVID-19 pandemic has demonstrated that airborne diseases are a significant threat to human individuals and societies. Especially during the initial phase, when the virus was unknown, several precautionary measurements were considered and introduced by governments all over the world. At that point in time, the effectiveness of the different measurements was entirely unknown and based only on assumptions. While this generally changes over time when scientists gather knowledge about the disease, the essential question of how high the probability of getting infected is at different places remains unanswered. This issue can best be addressed by representing dynamic human behaviour in different spatial domains. Agent-based modelling allows to computationally simulate this behaviour. Therefore, this report focuses on the combination of simulations of human behaviour in public spaces and scientific transmission parameters of airborne diseases to conclude site-specific infection probabilities. The extracted knowledge can be used to develop suitable measurements for specific places. Suitable and target-oriented measurements are crucial to discover the balance between enforcing the necessary restrictions and maintaining an active society and economy. That supports the prevention of human and economic losses. At the end of the project, the proposed model can be helpful to estimate the infected occupants. It allows to formulate a general index of transmission for the chosen application cases. In the future, indexes based on the proposed modelling framework could support decision-makers.

Keywords:

Airborne disease, agent-based modelling, transmission model, SARS-COVID-19

Preface

This project is written as the final project (master's thesis) with 30 ECTS for the MSc. Risk and Safety Management programme at Aalborg University in Esbjerg. It bases on the current versions of the university's curriculum and the learning objectives.

The main purpose of the project is to develop and assess an agent-based model to estimate transmission rates for airborne diseases based on simulated agent movement. The description of the model follows the ODD protocol.

The thesis offers a new modelling framework to evaluate disease transmission rates at specific locations considering their characteristics. It addresses other modellers in the same field of study and those developing disease control concepts for specific places.

All figures and tables are created by the author. If the contained information originates from a specific source, the phrase "based on [source]" is added to the description. Screenshots are indicated by "Screenshot [computer program]".

Members of Aalborg University can access all simulation files and the connected results following this LINK (AAU SharePoint).

Acknowledgements

The initial idea to write my thesis on disease transmission rates of airborne diseases and many other good ideas and problem solutions were introduced to me by José Guadalupe Rangel-Ramirez. He was my main contact person, and his influence in improving the model and the whole report cannot be overestimated. Thank you so much for your endless support, during the final project and the entire RISK programme, José!

Additionally, I am thankful to my brother for the feedback he provided. The comments from your external perspective and other scientific background have been valuable while identifying and eliminating indistinct formulations and missing explanations.

Acronyms

The acronyms are, additionally, explained when first used.

ABM	Agent-based modelling / Agent-based models
CDC	Centers for Disease Control and Prevention
COVID-19	Coronavirus SARS-CoV-2
csv	Comma separated values
E(-agent)	Epidemical status: Exposed (agent)
EC	European Commission
f	Female
I-agent	Agent with the epidemical status Infectious
L-agent	Agent with the epidemical status Latent
m	Male
nr	Number
R-agent	Agent with the epidemical status Recovered
S-agent	Agent with the epidemical status Susceptible (agent)
USD	United States Dollar (currency of the United States of America)
V-agent	Agent with the epidemical status Vaccinated
VI-agent	Agent with the epidemical status Vaccinated and Infected
VS-agent	Agent with the epidemical status Vaccinated and Susceptible
WHO	World Health Organization
WoS	Web of Science (www.webofscience.com)

List of contents

1 Executive summary

This master's thesis focuses on a modelling framework to estimate transmission rates of airborne diseases in public spaces without being limited to a specific disease. Airborne diseases are all diseases that spread through the air. Public spaces are defined by their accessibility to the general public. In addition, the general public is one of the four stakeholder groups – the remaining ones are authorities, the healthcare system, and companies. All of them have specific responsibilities and interests.

Regarding the model creation, previous research, identified through cluster analysis, based their transmission estimates on three different approaches: simulating the contacts, distances between the agents, or their movement. All approaches lead to a number and/or duration of contacts between susceptible agents (S-agents), who can become infected, and infectious agents (I-agents), who can infect others. The findings of this literature are considered for creating the model of this report.

The model itself splits the disease transmission process into two infection mechanisms: active and passive. The active infection mechanism represents the transmission from person to person. The passive one models the contact-free transmission through the spread of contagious particles that remain in the environment, after an I-agent comes across. As in the literature, the active infection mechanism relies on the number of contacts an S-agent has with I-agents. Other than these two epidemical statuses, the agents can be recovered (immune to the disease), and agents of all statuses can be vaccinated and/or wearing a face mask. The number of contacts is received by counting how many times each S-agent crosses the infection areas around I-agents. The value added to the contacts depends on how close the agents get to each other. That can be related to casual, intermediate, or close contact. The value is adapted if the I-agent is wearing a face mask. Afterwards, the active infection probability is calculated based on the number of contacts. The passive infection probability is influenced by the concentration of infected agents in the environment, the air exchange rate, and the individual agent's exposure time. The active and passive infection probabilities are summed up and reduced if the agent is vaccinated and/or wearing a face mask. The S-agents become infected with the calculated (reduced) probability. The number of new infected agents is used to compare the different environments and scenarios.

In addition to the initial minimal 5 x 5 m model with 20 agents, a public space 20 x 20 m model with 200 agents and with and without obstacles and three scenarios, including waiting behaviour, changing demographics, and including group movement, are created. The differences in the number of new infected agents indicate that higher agent concentrations, group movement, and waiting behaviour increase disease transmission, while older populations reduce it. The observed connections are plausible as the agents' immune system is not considered. The consideration of those could be a potential improvement of the proposed model. Other attributes that could be added, too. The plausibility of the passive infection mechanism is not proven because it remains at the lower limit value of 5 % for all the applied attributes, especially the short simulation times.

All in all, the created model fulfils the main objective of being functional and flexible enough to fit different environments and scenarios. However, the model leaves room for further improvements and larger-scale simulations using more powerful computers.

2 Introduction

2.1 Relevance

Diseases accompany humankind on its development to modern civilisations since ancient ages. The first proven epidemic occurred in a prehistoric village in China about 5,000 years ago, causing the death of all inhabitants (Jarus, 2021). Approximately 2,500 years later, the Athenian Plague of 430 B.C. spread from Ethiopia to Egypt and Greece with a lethal outcome for more than 25 % of the Athenians (Huremović, 2019). In the medieval ages, the black death killed between 30 % and 50 % of the European population within four years (DeWitte, 2014). Several hundreds of years later, the Spanish Flu (1918-1919, 20-50 million deaths), the Asian Flu (1957-1958, 1-4 million deaths), and the Hong Kong Flu (1968, 1-4 million deaths) were the most severe influenza pandemics in the 20th century (WHO, n.d.).

Today and in the previous two years, the world is facing a new pandemic: COVID-19. Originated in China, the virus started spreading globally in the first quarter of 2020 (Katella, 2021). The development of the total number of confirmed cases and deaths as of the 29th of December 2021 is presented in Figure 1.

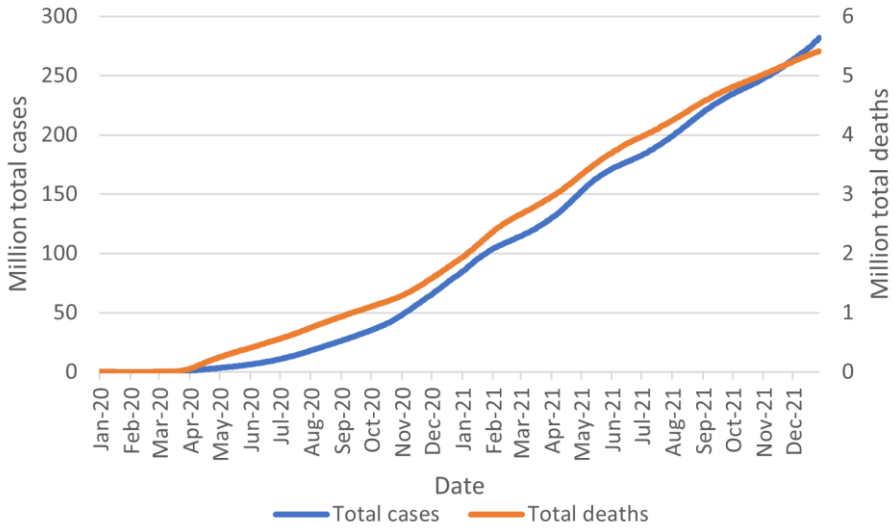

Figure 1 - COVID-19 - Total cases and deaths by date, based on (WHO, 2021a)

By the end of December 2021, more than 280 million people are or were infected, and more than 5.4 million people died with or caused by COVID-19. These numbers and the countermeasures have omnipresent economic and psychological impacts on societies all over the world. The estimated global economic impact is a deficit of 4.5 % gross domestic product (GDP), or almost three trillion USD, for 2020 only (Szmigiera, 2021). While the total economic loss is increasing as the pandemic continues, the psychological impact cannot be expressed by a single value. The precise psychologic reaction is individual for each person. Generally, the fear of becoming infected and suffering under severe symptoms, combined with loneliness caused by social distancing and lockdown measures, reduces subjective wellbeing and even results in depressions for some individuals (Serafini et al., 2020).

To limit the economic losses and increase subjective wellbeing, while – at the same time, effectively fighting a pandemic, the introduced measures must be as efficient as possible, rejecting costly measures with only minor benefits.

2.2 Audience

This report addresses local authorities, companies, and other researchers in the same field of study providing a framework to model and estimate the transmission of airborne diseases at specific indoor locations. A detailed project description follows in the next sub-chapter. How the different parts of the audience can benefit from the findings of this report is explained below.

During a pandemic, the authorities are responsible for introducing countermeasures. At the local level, these measures may include restrictions for specific locations. The local authorities can apply the introduced model to evaluate the appropriateness of potential restrictions. If the authorities demand the companies to develop their own transmission reduction concepts for their facilities, the companies can as well use the model in the decision-making process. The same is the case for the authorities in the case of publicly owned buildings.

Furthermore, the proposed model can be used as a starting point for future development by other researchers. The first ideas of potential future research topics are in chapter 8.2.

2.3 Scope

2.3.1 Problem statement

If the transmission rate of a disease, which may lead to severe symptoms, reaches a limit, there is a need to enforce preventive measurements. As described in chapter 2.1, all measurements have an impact on society. Therefore, the decision-makers should consider all foreseeable consequences of the discussed measurement options to weigh up potential benefits and losses before making the decision. Especially during a pandemic, it is essential to introduce a set of well-coordinated international, national, regional, and local measurements, concerning the unequal disease development at different places to form efficient policies with increased acceptance among the general public.

Besides good cooperation of the involved decision-makers, multi-level policies require sufficient information on every level. The decision-makers must have as much information available as possible, to make the best possible decision. In the context of diseases, the relevant information is, e. g., connected to the severity of symptoms, risk groups, and transmission rates based on transmission procedures and the connected infection probabilities.

One way of gathering this information is to use computer simulations. These simulations have the advantage that they can provide quick answers on the expected effects of policy changes even before these are applied in the real world. On the contrary, the preciseness of simulations heavily depends on the appropriateness of the created models.

2.3.2 Problem question

This report focuses on the final part of the problem statement above. In short, it is expressed by the problem question below:

How can the influence, local conditions such as demographics, human behaviour, and room conditions have on the transmissibility of airborne diseases in public indoor spaces be quantified based on agent-based modelling computer simulations?

The question consists of the five main aspects local conditions, transmissibility, airborne diseases, public indoor spaces, and agent-based modelling. The local conditions refer to all attributes connected to the specific system to be modelled. As mentioned in the question, local conditions divide into demographics, human behaviour, and room specifics (e. g. room dimensions and ventilation). Transmissibility is understood as the process of becoming infected – the central aspect of the model. As the name implies, airborne diseases are defined by being transmitted through the air. Public indoor spaces are all places that are publicly accessible. The modelling type applied is called agent-based modelling (ABM). Within this approach, the focus is on the individual occupant, the agent.

Precise information on the different aspects is provided where they are applied. The connected chapters follow the list below:

- Local conditions: Hierarchy Figure 11
- Transmission Procedure Figure 14, chapter 6.3.3
- Airborne diseases Definition chapter 4.1
- Public indoor spaces Definition chapter 4.1
- Agent-based modelling Definition chapter 5.1.1, Application chapter 6

2.3.3 Objectives

2.3.3.1 Main objectives

Two main objectives break down the problem question into achievable tasks. Consequently, meeting the objectives results in answers to the problem question. The objectives are stated and explained below.

Objective 1: Develop a model to simulate and estimate transmission rates for airborne diseases using ABM.

The first objective addresses the model creation, its aim and approach. The model is to be built up as an ABM, following the connected procedure and characteristics. To analyse the influence of local conditions, the model must result in comparable outcomes representing the transmission rates.

Objective 2: Test the model analysing relevant inconstant local conditions regarding their influence on the transmission rate.

After a basic version of the model is created and functional, a set of relevant local conditions is implemented in the model and further analysed. The analysation results in descriptions of the observed influence of the individual local conditions. Then, these influences are evaluated regarding their plausibility as a next step to confirm the model's functionality.

2.3.3.2 Sub-objectives

Further developing the main objectives stated above, sub-objectives are formed below, dividing the tasks into smaller work packages. The sub-objectives are obtained from the connected main objectives by asking questions like “What is meant by that?” and “What needs to be done to fulfil that task?”.

Sub-objective 1.1: Analyse and build upon the current state of the art in ABM of airborne diseases.

While creating a new ABM for estimating the transmission of airborne diseases, previous work with similar objectives should be considered to start from the current state of the art. Building upon this starting point, the model benefits from previous modelling ideas and lessons learned. Furthermore, this approach ensures advancing research rather than duplicating past models.

Sub-objective 1.2: Account for and combine active and passive infection mechanisms.

In the development of the model, it grows and becomes more and more detailed. One example is separating the general infection procedure into an active and passive infection path. The active infection path accounts for disease transmission from person to person (agent to agent). In addition, the passive infection path considers the infection of a person caused by disease particles remaining in the air after an I-agent passed by.

Sub-objective 1.3: Ensure sufficient flexibility of the created model, so that it is adaptable to fit other locations and scenarios.

The aim is not to create a model fixed to a specific location or scenario. It should be flexible enough to fit different locations and scenarios. Hence and whenever possible, input values must be assigned to defined variables and exchangeable if needed.

Sub-objective 1.4: Provide understandable descriptions of all modelling and simulation steps to make them accessible and reproducible, especially in the context of future work in this field of research.

All aspects of the model must be understandable by the reader to ensure adaptability by other modellers. The structure of the model, containing all sub-models and the underlying ideas such as geometrical-mathematical relationships and equations, must be described extensively supported by graphical representations.

Sub-objective 2.1: Select a set of the identified inconstant local conditions to be further analysed in this report.

Considering the limitations of this report, it is not possible to analyse the impact of all local conditions. These must be limited to a set of the most relevant ones. For a local condition to be relevant, it must be modellable and expected to have a detectable influence on the transmission rate.

Sub-objective 2.2: Highlight the opportunities for future research connected to both, the inconstant local conditions further analysed, and the ones not further analysed in this report.

Connected to the selection of local conditions and the limitations, this report leaves room for further research. As the transmission of airborne diseases currently is a hot topic, promising extensions of this report should be highlighted, motivating others to continue research in this field of study.

2.3.4 Limitations

Since this report is the outcome of a student project, it bases on a learning-by-doing process. That is especially the case for the programming sections. The code is written and corrected step by step, resulting in a functional version that is not necessarily the most efficient possible.

The model itself comprises several agent (Figure 10) and environment attributes (Figure 11) that are expected to have a significant influence on the transmission rate. However, these lists are not all-encompassing, and limited to what is modellable with reasonable effort. Moreover, the modelled attributes are often simplified or summarised in ranges. The agent profiles, e. g., define the agents by their gender and age within spans of up to ten years and ends with the category older than 70 years. Generally, the simplifications rely on plausible assumptions but are not scientifically proven. The model is, therefore, to be updated with developing scientific knowledge. Regarding the disease-specific input, it is to note that the assigned values are not based on real diseases and must be replaced in case a specific disease is to be evaluated. The model representation (chapter 6) contains further and more detailed limitations and assumptions.

Explicitly implemented transmission control measures are vaccinations and the usage of face masks – other measures are not considered directly. Neither considered is the legal background for applying measures as it varies from nation to nation. Generally and If needed, the responsible authorities change the relevant laws to implement the necessary measures.

The extent of this report is mainly limited by time constraints. In connection with the available computing power, the available time limits the number of scenarios that are analysed. Concerning simulation times of up to 5.5 h per repetition, the number of scenarios is set to three.

3 Methodology

In this report, the chapters are not entirely in the same order as the project work is done. To support the readers understanding of the work process and improve reproducibility, Figure 2 presents the work packages in chronological order. The work packages are ordered on the arrow and counted through by the circled numbers above. Below the arrow, the connected chapters in this report are listed using their numbers. With the exemption of the sub-chapters 6.4 and 6.5, which are connected to the model testing, the listed chapter numbers refer to the complete chapters containing all sub-chapters.

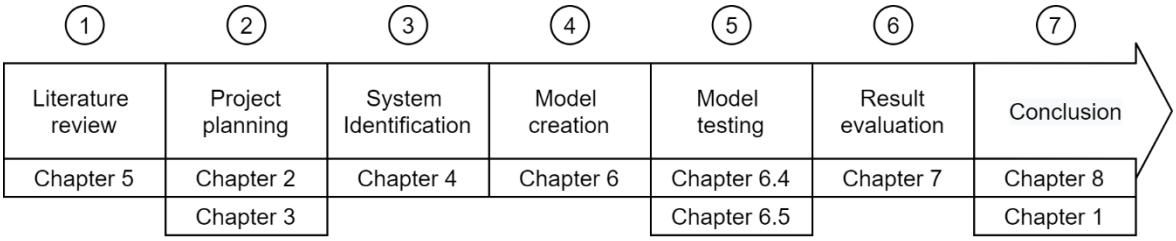

Figure 2 - Project methodology

The project work starts with a literature review (chapter 5) to analyse the relevance and the current state of the art of disease transmission modelling. Based on the findings, the project is planned formulating what is to be considered (chapter 2). That includes the objectives, especially the problem question, the limitations, and the target audience. This methodology also belongs to the same step as it contains the planned structure of the report. It is updated with the further development of the project.

When the topic and initial structure are known, the third step is to understand the real system. That is done in the system identification (chapter 4), focussing on airborne diseases, public accessible spaces as a delimitation of private accessible domains, and the involved stakeholders.

The model, which is the centre of this report, is created in the fourth work package, starting with a minimal model to work with. Chapter 6 explains the detailed creation process, connected assumptions, and functionality. When the model appears to be functional, it is initially validated by examining the plausibility of the outcomes for extreme cases. For example, if there are no I- or no S-agents in the model, nobody can become infected. After the model's results for all these cases are plausible, it is expanded to the public space model (chapter 6.4), and a set of scenarios is created (chapter 6.5) in the fifth work package. While creating the public space model and the scenarios, specific attributes are changed to test, whether the model is sensitive to these changes, and analyse the connections.

Afterwards, in step 6, the results of all model versions and scenarios are evaluated, first individually, then comparatively (chapter 7). Lastly, the comparison and all findings of the other chapters are referred back to the problem question and objectives in the conclusion (chapter 8.1). Furthermore, this work package contains a brief outlook into promising future extensions to this report (chapter 8.2) and the executive summary (chapter 1) as a short version of the entire paper.

4 System Identification

4.1 Airborne diseases in public and private accessible domains

Dealing with airborne diseases requires a common understanding of the term itself. The part airborne refers to the transmission type. Airborne diseases are in most cases spread by infectious aerosols. Other options are skin flakes and fungal spores in the air or physical contact with contagious surfaces or persons (Aliabadi et al., 2011). Generally, large droplets from the nose and mouth are not spread more than 2 m from the source, but smaller ones may evaporate and remain in the air, spreading in a larger area (Flores and Cohen, 2014). A disease is "any harmful deviation from the normal structural or functional state of an organism, generally associated with certain signs and symptoms" and is also referred to as illness (Burrows et al., 2020).

The models created for this report focus on public accessible indoor environments. Publicly accessible means that the building or, at least, the relevant parts are open to the general public. Typical examples are shopping malls, exhibition halls, and hospitals. At these places, the people present are, generally, not familiar with each other and follow a specific interest, e. g. buy goods or receive medical treatment. The occupation at public domains changes frequently and often continuously. In the shopping mall example, the customers only visit the centre and its stores for short periods and are followed by the next customers. Contrarily, private accessible domains are open only to a specific, generally small, group of persons. These can be domestic buildings, companies without customer traffic, or other areas with restricted access like military zones. There, the people are more familiar with each other, and the population is more or less constant.

4.2 Stakeholder Identification

A stakeholder is a person or group interested in a system or, especially, upcoming decisions. Following this definition, several different stakeholders can be identified due to their interest in reducing the risks connected to the spread of airborne diseases and/or potential measurements. To highlight their relations, common and opposing interests, the stakeholders are grouped into the four categories shown in Figure 3.

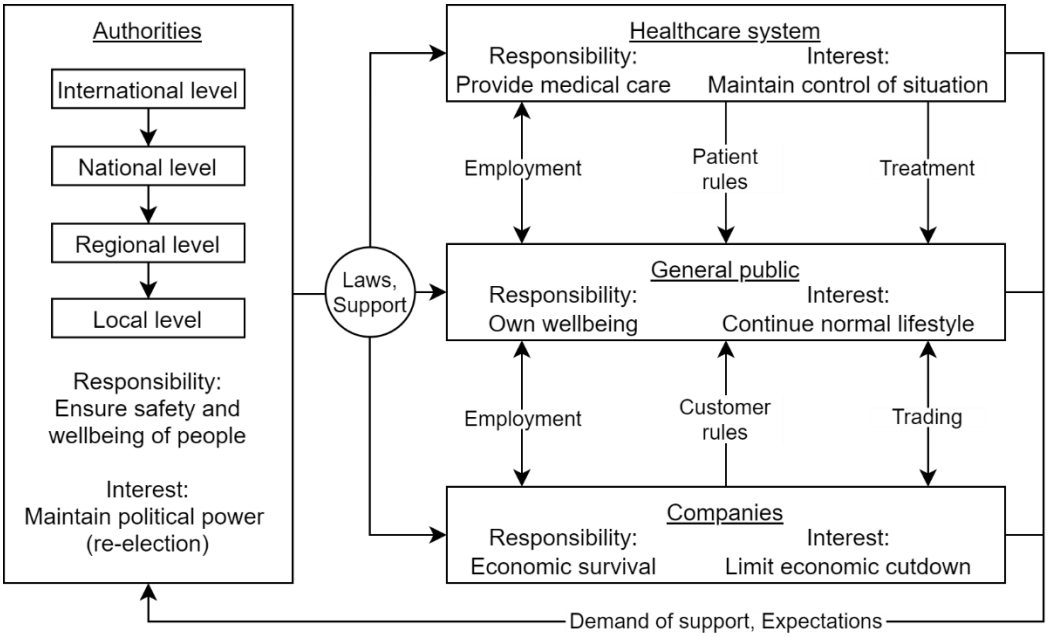

Figure 3 - Stakeholder interactions

The current state of the system is mainly dependent on laws passed by the authorities since these define the legal interactions of all stakeholders. There are four levels of authority: international, national, regional, and local. The distribution of tasks among the different levels depends on the involvement in international organisations and the centralisation degree of the analysed nation. Generally, all authorities have to follow laws passed by relevant higher-level authorities and formulate additional laws for lower-level ones and their people. While considering new or the enforcement of existing laws, the authorities must fulfil their first responsibility to ensure the safety and wellbeing of their people. At highest priority, that means survival and healthiness but should also include financial and social aspects. Besides this responsibility, the authorities, respectively the elected politicians, follow a personal interest of maintaining their political power and becoming re-elected. Especially before new elections, unpopular decisions are, therefore, less likely. Most disease control measures are designed to limit contact by reducing social activities. In some cases, that may include lockdowns of entire sectors, e. g., those offering free-time activities.

From the economy's perspective, lockdowns should be avoided since all private companies must make a profit to survive. Dependent on the sector, the lockdown risk varies from company to company. The ones selling essential goods like groceries and others without customer traffic face comparatively low risks, while companies in the free time sector (e. g., nightclubs, sports stadiums, and festivals) face higher risks. Some other companies may even benefit from closed down concurrence. If local stores are closed or opened with restricted access, online delivery services likely make more profit. Regardless of all the differences within this diverse stakeholder category, all of them share the interest of limiting their economic losses caused by an epidemic and the countermeasures. The connections to the general public are ambivalently caused by employment and customer relations. For employment, the relationship is reciprocal, because the employer and employees influence each other. The companies depend on the workforce of the employees and, therefore, their workability (healthiness). The employees, on the other hand, rely on their payment and the economic success of the company. The customers are not in such a close relationship with a single company as there are typically several companies to choose from to exchange their money for goods or services. In this relation, the companies define a set of rules the customers have to follow. These rules can be initiated by the companies or fulfilling requirements stated by the authorities.

The general public is the centre of the system as it consists of all the people who may become infected and infect others. As seen in the figure, the general public is exposed to several laws and rules from all sides. All rules conflict with the fundamental interest of continuing a normal lifestyle without restrictions. The main responsibility of each individuum is to maintain its wellbeing. Within social groups like families, that is related to the groups' wellbeing. To meet this responsibility, the people follow the given rules as much as they expect them to be efficient and necessary, or they fear punishment. Furthermore, the overall obedience depends on the trust level of the authorities. The central connection to the healthcare system as the fourth stakeholder category is caused by the treatment of infected people with severe symptoms.

The healthcare system provides medical care based on patient rules that can be self-initiated or created by applying laws. All kinds of relevant medical institutions are grouped in this category. Therefore, it comprises large-scale organisations like state hospitals and

small facilities like doctor's offices. They all share the common interest of maintaining control of the situation, so that the demand for treatment does not exceed their capacities. Privately owned institutions additionally share the companies' responsibilities and interests stated above. As the companies, the healthcare system depends on the workforce of their employees, who are a part of the general public. Hence, maintaining a functional healthcare system is only possible if a critical quota of healthcare workers remains able to work.

The healthcare system, the general public, and the companies all have specific expectations on how the authorities should manage the situation. Considering the difficulties the individual stakeholders have, they ask the authorities for, typically financial, support. How the authorities react to this demand and distribute the available resources, is critical during a pandemic.

5 Theoretical framework

5.1 Literature review

The literature review is conducted as an initial step to create an overview of the current state of the art, representing the present understanding of agent-based-modelling (ABM) in general, and specifically for the modelling of disease transmission. The reference selection process is conducted iteratively by introducing, analysing, and improving search queries and their results. It starts with an initial search query containing keywords of the three categories disease-related, ABM related, and methods/tools related. The query is created so that at least one keyword per category is found for a piece of literature to be relevant, resulting in 320 search results. The keywords connected to the categories, the search queries, and the number of results are in the appendix (chapter 12.1) with explanations for modifications. The found keywords and other repeatedly found words (10+ occurrences in title and abstract) are analysed, grouped, and in some cases disregarded before creating clusters out of them. Analysing the clusters and their visualisation supports identifying irrelevant topics among the search results. If that is the case, the search query is adapted accordingly, and the process starts again with the new query. That continues until no irrelevant topics can be identified through the clusters. Then, the remaining references are checked manually to make the final selection of relevant literature to be included in this project. The circular approach of this process is represented in Figure 4.

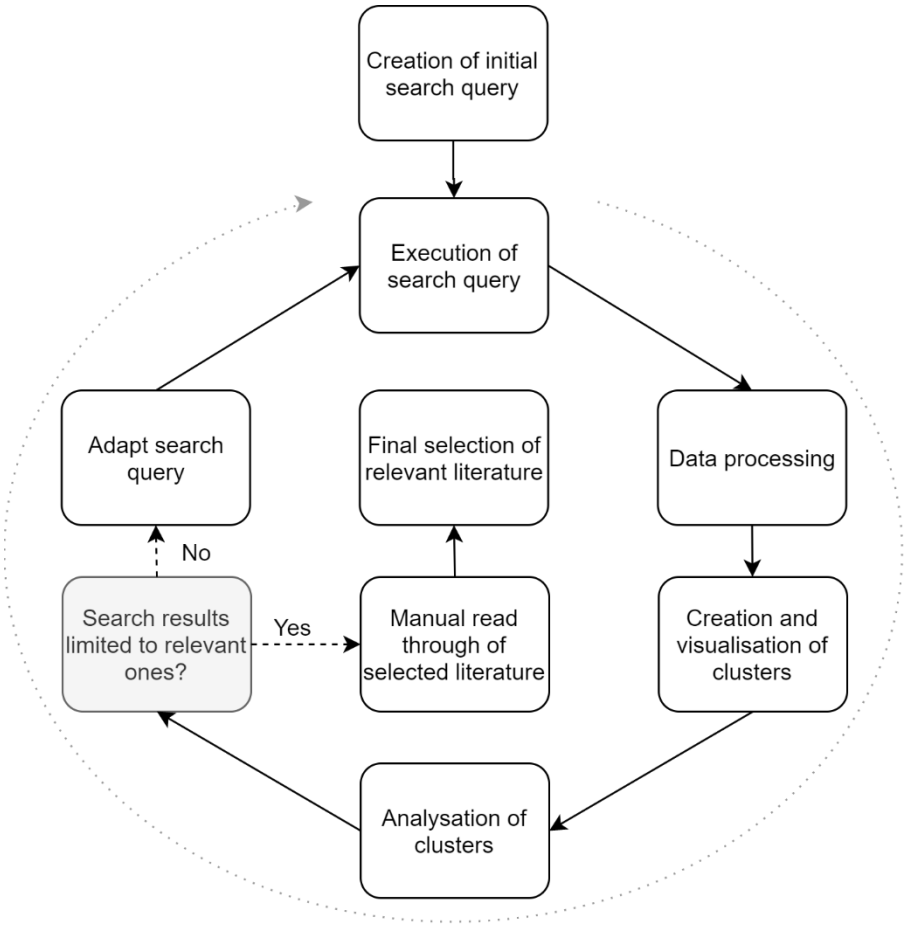

Figure 4 - Iterative process of reviewing literature

The cluster analysis, the key element for eliminating irrelevant search results, provides an overview of the connections between keywords used in the analysed literature. As an example, Figure 5 shows the clusters for the final search query.

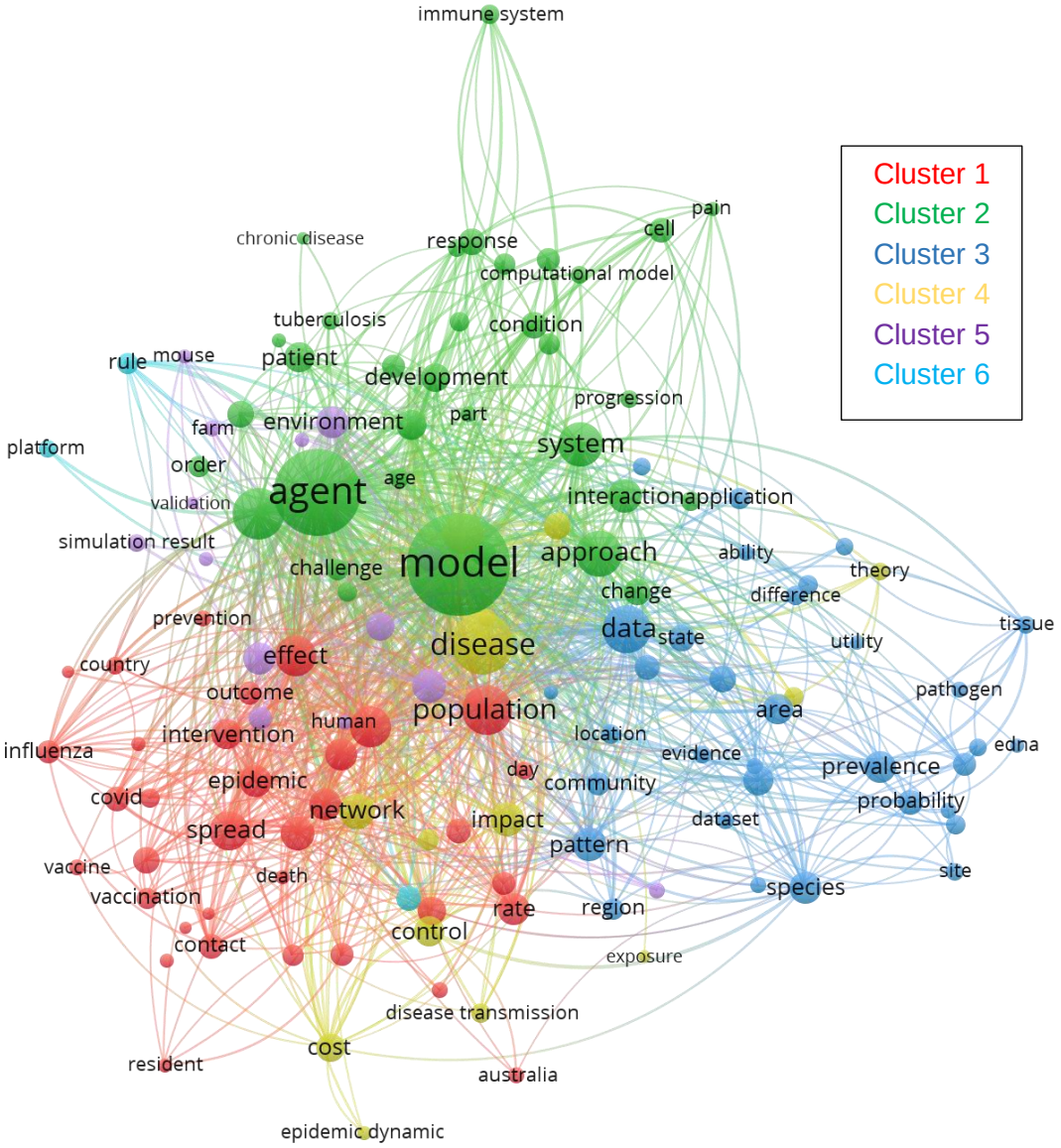

Figure 5 - Cluster network for the final search query, Screenshot WoS

Here, six clusters are formed and highlighted using different colours. Cluster 1 (red) mainly combines keywords in the field of transmission of diseases and connected control measures (e. g. contact, outbreak, strategy, vaccination, and control). Cluster 2 (green) mainly contains modelling keywords (e. g. model, agent, system, mechanism, and understanding), Cluster 3 (dark blue) model input data keywords (e. g. area, data, knowledge, location, and occupancy), Cluster 4 (yellow) ABM keywords (e. g. person, influence, environment, epidemic dynamic, and validation), Cluster 5 (purple) the decision-making keywords (e. g. assumption, change, risk, information, and value), and Cluster 6 (light blue) unrelated keywords (infectious disease, platform, and rule). However, these are not exclusive categories – some keywords are not related to the topics in each category.

The most relevant literature for this report is connected to the combination of clusters 1 and 2 for general disease transmission modelling. Respectively, clusters 4 and 2 are essential for ABM of disease transmission in specific. Cluster 4 alone provides information on ABM

techniques and, therefore, a basis to start with. The main findings are presented in the following sub-chapter 5.1.1. The current state of the art in disease transmission modelling is in sub-chapter 5.1.2.

5.1.1 Agent-based modelling

"*Agent-based modelling is a computational approach in which agents with a specified set of characteristics interact with each other and with their environment according to predefined rules*" (Tracy et al., 2018).

This definition combines all essential elements of ABM – the agents as the centre of the model ("agents […] interact"), the mathematical simulation ("computational approach"), the influence of the environment ("agents […] interact with […] their environment"), and the interaction restrictions ("predefined rules").

Depending on the system, an agent can be a single person, a group of persons (e. g. a family), but also nonhuman individuals such as a specific animal population. It is essential to model the agents in a degree of detail, sufficient as far as relevant for the analysed system. If the model considers human agents, it is not necessary to draw a full picture of each agent, e. g., containing hair and eye colours. Moreover, it should be avoided to include unnecessary details since that can result in difficulties in testing and executing the model (Auchincloss and Garcia, 2015).

The influence of the environment addresses all system parameters that must be considered in the model but are not a part of the agents themselves. As for the agents, detailed information about the environment is only needed, as far as it influences the outcome of the simulations. In this connection, influences can be direct and indirect (Grimm et al., 2006). Therefore, direct influences should be considered as starting points of possibly long chains of indirect consequences.

The interaction restrictions limit the theoretically endless interaction possibilities to a smaller set of possible interactions considered in the model. Typically, complete information on how likely which specific interaction and its connected outcomes are is not available, so the interaction restrictions must be based on what is modellable considering limited information and common sense (Auchincloss and Garcia, 2015).

The mathematical simulations are the conceptual backbone of the ABM approach. Following the agents' characteristics, their environment, and interactions, the model is executed, concerning a logical-mathematical understanding of the context. Therefore, this aspect of ABM refers to creating mathematical equations and/or programming code used to estimate the risks connected to specific scenarios. Since this element represents the main modelling ideas of the creators, it is to be described and explained precisely to ensure accessibility for the audience (Tracy et al., 2018).

Concluding the information mentioned above, ABM can be used to describe complex systems. However, this implies that the created models may be complex, too. Therefore, it is critical to describe the models in a comprehensive and well-structured manner to increase understandability. For this purpose, 28 international researchers developed the ODD protocol described below (Grimm et al., 2006). The ODD protocol is divided into three blocks, two of them are further sub-divided into elements (Figure 6).

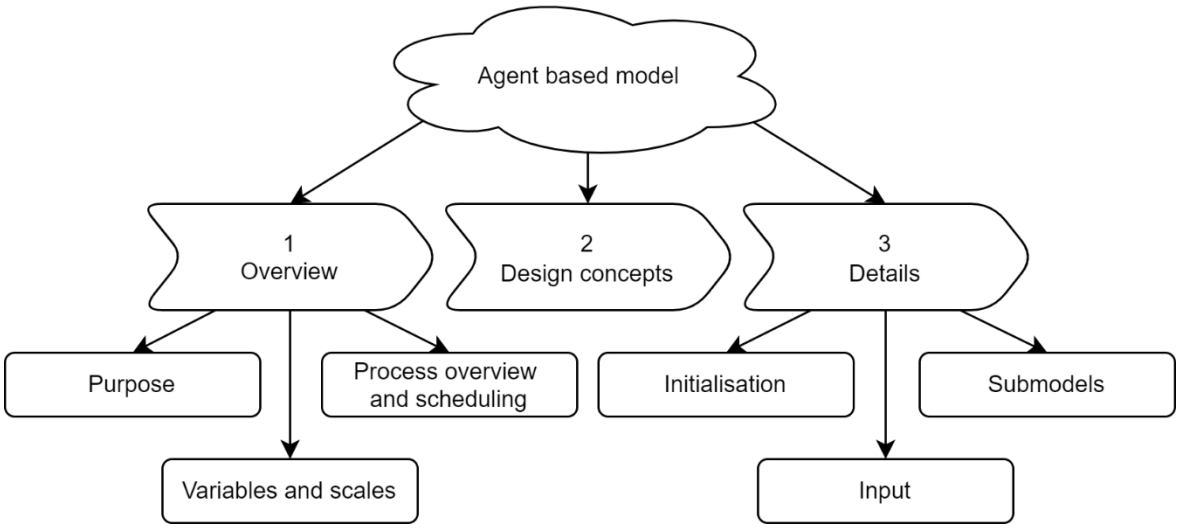

Figure 6 - Overview ODD protocol, based on (Grimm et al., 2006)

The first block **Overview** contains the basic information about the ABM, starting with the purpose of the model. Connected to the purpose is a brief explanation of why the analysed system is represented by using an ABM instead of other modelling approaches. Secondly, the modelled variables are explained, starting with the low-level variables. Low-level variables are the initial input of the model and influence the higher-level variables explained afterwards. The term scales refers to the dimensions of variables and the model in general, e. g., the length of time steps within the simulation. The process overview contains information about the modelled interactions between the agents and their environment, while the scheduling part puts them in the right order and highlights connections between them.

Block 2 addresses the particular design of the model (**design concepts**). This individuality is problematic to describe in general. Therefore, (Grimm et al., 2006) provides a set of questions to pick from for this part of the model description. Answering the questions provides insights into the modellers understanding of the system, especially the agents. The questions cover the sections emergence (pre-defined versus simulated behaviour), adaption (supportive/preventive behaviour), fitness (target-oriented behaviour), prediction (foreseeing decision outcomes), interaction (mutual influencing), sensing (environment perception), stochasticity (input values variability), collectives (agent groups), and observation (result perspective and validation). As far as relevant for the model of this project, more information about the second block is added when applied in chapter 6.2.

Lastly, as the name implies, the model is described with respect to all **details**. In the initialisation element, the initial values of the variables are stated and explained. If applicable, it is done for all scenarios. Afterwards, the input element describes the origin of the input values. The submodels element revisits the processes from the process overview and scheduling element – adding detailed information, e. g., exact equations and programming code.

5.1.2 State of the art – Disease transmission modelling

The manual read-through as the last step of the literature review identifies eight relevant pieces of literature representing the current state of the art of disease transmission modelling. These are, in alphabetical order:

- A probabilistic model to evaluate the effectiveness of main solutions to COVID-19 spreading in university buildings according to proximity and time-based consolidated criteria (D'Orazio et al., 2021a)
- Agent-Based Modeling of the Spread of Influenza-Like Illness in an Emergency Department: A Simulation Study (Laskowski et al., 2011)
- Agent-Based Simulation Framework for Epidemic Forecasting during Hajj Seasons in Saudi Arabia (Alshammari et al., 2021)
- An agent-based approach for modeling dynamics of contagious disease spread (Perez and Dragicevic, 2009)
- An agent-based model to evaluate the COVID-19 transmission risks in facilities (Cuevas, 2020)
- Investigating transmission dynamics of influenza in a public indoor venue: An agent-based modeling approach (Zhou et al., 2021)
- Predicting Self-Initiated Preventive Behavior Against Epidemics with an Agent-Based Relative Agreement Model (Mao, 2015)
- Sustainable and resilient strategies for touristic cities against COVID-19: An agent-based approach (D'Orazio et al., 2021b)

As all these focus on similar objectives in connection to ABM of disease transmission, the agent characterisations and the infection simulation approaches are also similar. In all the references, individual human agents are modelled with respect to their epidemical status. The complexity of the applied epidemical models varies from binary susceptible-infectious models to complex models containing up to four different epidemical statuses. Hereof, the *SEIR*-model is the most common. If grouped with the almost identical *SLIR*-model, it is the basis for half of the eight references (3+1). The underlying idea that a non-infected and non-immune (S-) agent may become infected after being in contact with an I-agent is the same in all references. The used terms are also mostly the same.

The complete epidemical process, the most complex previously used approach, consists of four statuses. After being infected, the S-agent is exposed to the disease. In this phase, the agent is infected but not yet infectious. The disease develops in the agent until he becomes infectious himself. Since this phase is also called the latent period, (Mao, 2015) uses latent as a status keyword instead of exposed. The meaning is practically the same. If the simulated period is long enough, exposed agents become infectious themselves, and may infect other S-agents. Afterwards, the disease dissolves in each agent. In the epidemic models, the status is named after the optimistic outcome that the agent recovers. Other outcomes are possible, including long-term medical consequences and death. Dependent on the specific disease, recovered agents (R-agents) may be immune. Figure 7 contains an overview of the above-explained. Less complex models do not differentiate between the exposed and infectious status and/or do not consider the recovered status.

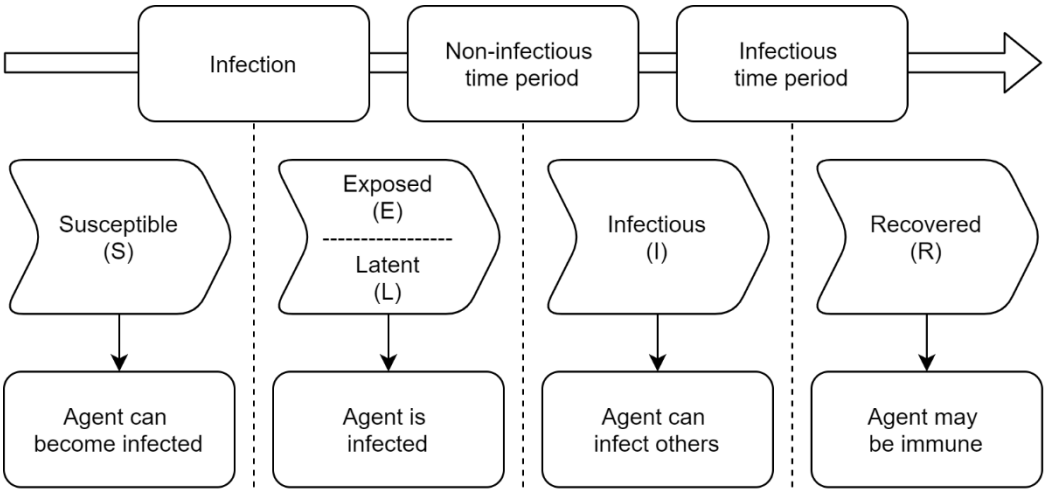

Figure 7 - Epidemic models in state of the art literature

The further degree of detail in the agent modelling ranges from basic combinations, e. g., only adding movement rules or group assignment, to comprehensive descriptions of demographical, behavioural (general and preventive), and biological attributes. Demographic attributes are connected to the age, gender, and origin of the agents. Behavioural attributes relate to movement patterns and the adaptation of preventive measurements (mainly wearing face masks). Biological attributes consider the development of symptoms in infected agents. In addition, (Mao, 2015) features the attitude towards preventive behaviour as a communicational attribute. The different agent attribute categories and their belonging attributes are shown in Figure 8.

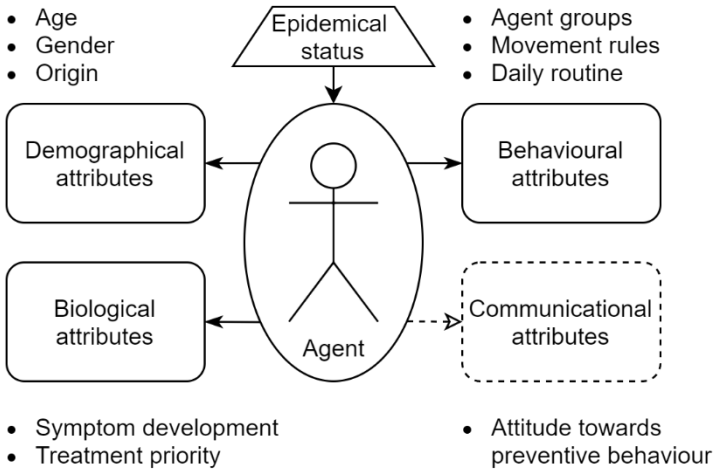

Figure 8 - Agent attributes in state of the art literature

The second primary aspect of ABM for disease transmission is the simulation part. Here, the different modellers selected different approaches concerning their objectives. Three basic approaches are identified and represented in Figure 9. All three simulation approaches start with assigning the attributes to the agents and continue with a procedure to estimate whether S-agents get in contact with I-agents. If that is the case, S-agents become infected with a certain probability which might change their epidemical status. For iterative simulations, a simulation parameter changes, and the last two steps are repeated for each simulation step. The simulation ends with a final epidemical status for each agent. The detailed procedure is different in the approaches.

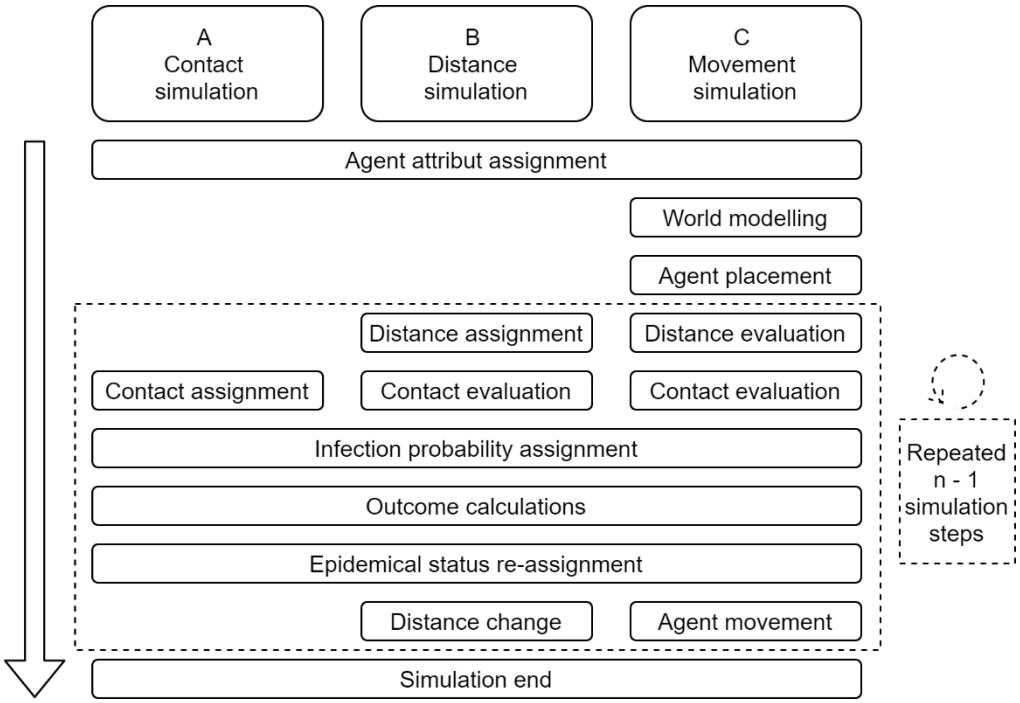

Figure 9 - Infection simulation approaches in state of the art literature

Simulation approach A (contact simulation) is based on direct contact assignment to each agent. This assignment contains contact calculations, typically based on behaviour observed in the real world. In the next step, the infection probability for S-agents is assigned based on the number and duration of contacts with I-agents. Simulation approach B (distance simulation) assigns distances between the agents to evaluate the contacts instead of calculating them directly. The distances are re-assigned after each simulation step, resulting in a certain infection probability if the distances are below a threshold. Simulation approach C (movement simulation) features a modelled world the agents are placed in. The positioning and simulated movement of the agents result in concluded distances between S- and I-agents. Evaluating these distances results in infection probabilities, and the epidemical statuses are re-assigned. Brief descriptions of the agent characterisation and simulation procedures of the individual state of the art literature are in the appendix (chapter 12.2).

Besides all similarities and common simulation approaches, there are also differences and particularities. As described above and marked by a dashed line in Figure 8, (Mao, 2015) focuses on the influence of personal attitude and discussion on the adaption of preventive behaviour and, in combination with the efficiency of preventive behaviour, the resulting changes on the overall disease transmission rate. The connected agent attributes are not considered in the rest of the reviewed literature. Based on several assumptions and scenarios, the journal article highlights the importance of a widespread positive attitude towards preventive behaviour among the general public to limit disease transmission. The other references add specific agent attributes in direct connection with their analysed cases, e. g., agent groups such as employees and customers in indoor venues (Zhou et al., 2021), patients and healthcare workers in emergency departments (Laskowski et al., 2011), and pilgrimage groups during the Hajj season (Alshammari et al., 2021). (Laskowski et al., 2011) furthermore expands the contact evaluation to a three-state concept: no contact, casual contact, and close contact, implying that closer contact results in a higher infection

probability. The conclusions and recommendations of the references considering preventive behaviour differ in detail but agree that the (proper) usage of (adequate) facemasks significantly reduces the transmission rates of airborne diseases (Cuevas, 2020; D'Orazio et al., 2021b, 2021a; Laskowski et al., 2011).

Comparing the available reference literature to the objectives and model of this report (chapters 2.3.3 and 6), the report addresses a previously unconsidered sub-topic of ABM of disease transmission. Previous models considering simulated agent movement are based on typical (observed) daily routines addressing the bigger picture but being less accurate regarding precise movement within a room. The framework proposed in this report furthermore combines a differentiation in active transmission based on a three-part octagon-shaped infection area and passive transmission based on the concentration of I-agents and several room-specific attributes – New ground in disease transmission modelling which is, especially within the current situation (see chapter 2.1), interesting and promising to look at.

5.2 Glossary and definitions

The keywords explained below (Table 1) may have several definitions in other contexts. The explanations represent the understanding of the keywords as used in this report.

Term	Definition
Agent-based modelling (ABM)	See chapter 5.1.1
(Airborne) disease	See chapter 4.1
Cluster analysis	An algorithm-based analysis tool for ordering data sets. The data points are grouped into categories (clusters) in a way that all data points within a cluster are more similar to each other than to the ones in other clusters. In this report, the term cluster analysis is limited to the analysis of potentially relevant literature using a computer software called VOSviewer.
Coronavirus SARS-CoV-2 (COVID-19)	COVID-19 is an infectious disease caused by the SARS-CoV-2 virus. Typically, infected people suffer only mild or moderate symptoms and do not need medical treatment. Some, especially older or chronologically ill people, experience serious symptoms or even die with or caused by the virus. (WHO, 2021)
Epidemic	An extraordinary high number of disease occurrences per time in a specific area or population group (CDC, 2012).
Epidemical status	Individual agent status in relation to the disease based on the epidemical model applied (see Figure 7 and explanation above).
Gross domestic product (GDP)	“The total of all value added created in an economy”, considering all goods and services produced in the economy minus the ones imported (EC, 2019).
Hajj	Annual religious Muslim pilgrimage to Makkah in Saudi Arabia (Alshammari et al., 2021).
Kite	A quadrangular flat shape with two pairs of adjacent sides. Both sides of a pair are equal in length (Pierce, 2021).
Octagon	A flat shape (polygon) with eight sides (Pierce, 2018).
Pandemic (influenza)	A globally spreading disease with no or only minor pre-existing immunity in the population. Symptoms may vary between mild and terminal. (WHO, 2019)
Pathfinder	An agent-based evacuation simulation software. More information can be retrieved from the publisher’s website (Thunderhead, 2019).
Public space	See chapter 4.1
Python	A programming language, in this report used to automate the simulation repetitions.
R	A programming language, in this report used to evaluate the output from Pathfinder into infection probabilities and outcomes.
Risk	The combination of consequences and their occurrence probabilities; if numeric, calculated as sum of the products of probabilities and consequences.

Table 1 - Glossary

6 Model representation

The following representation of the model is based on the ODD protocol described in (Grimm et al., 2006) and paraphrased in chapter 5.1.1, especially in Figure 6. The described model is the first minimal model used in the development process. Most of the modelled elements remain unchanged for the later public space model. Changed elements are highlighted in sub-chapter 6.4.

6.1 Overview

6.1.1 Purpose

The first purpose of the created model is to analyse the influence of local conditions on the transmission rates of airborne diseases, divided into active and passive infection mechanisms. More detailed information on the orientation and objectives of the model and this report, in general, can be found in chapter 2.3.

The ABM approach (described in chapter 5.1.1) is selected, because of its focus on individual agents. In disease transmission modelling, personal aspects such as age, vaccination status, and movement patterns influence the resulting individual infection probabilities. This assumption is plausible based on the findings of the reviewed literature (chapter 5.1.2). Furthermore, ABM can include all kinds of environmental parameters and their influence on the agents' behaviour through defined movement rules. In the creation process, the first rudimentary models are expanded gradually by adding more attributes resulting in more complex and precise models of the real world. The iterative process of improving the model combined with the optional adding of theoretically infinite agent and environment attributes is a powerful tool while researching beyond commonly known facts.

6.1.2 Variables and scales

The variables and scales also referred to as attributes, are divided into a hierarchy of up to seven levels and the categories agent attributes (Figure 10), environment attributes (Figure 11), and concluding attributes (Figure 12). The figures give an overview of the essential attributes used in the model. Therefore, some intermediate variables, especially fixed values, are not represented in the figures. In these cases, the left-out variables are considered in the explanation below the corresponding figure. More detailed information is in chapter 6.3.

In the three figures, the hierarchy levels are indicated by circled numbers next to the attributes. The attributes of the lowest level (level 1) are listed with their name, an abbreviation in round brackets, and the unit in square brackets. The ones at higher levels additionally feature the lower-level attributes they are based on in curly brackets.

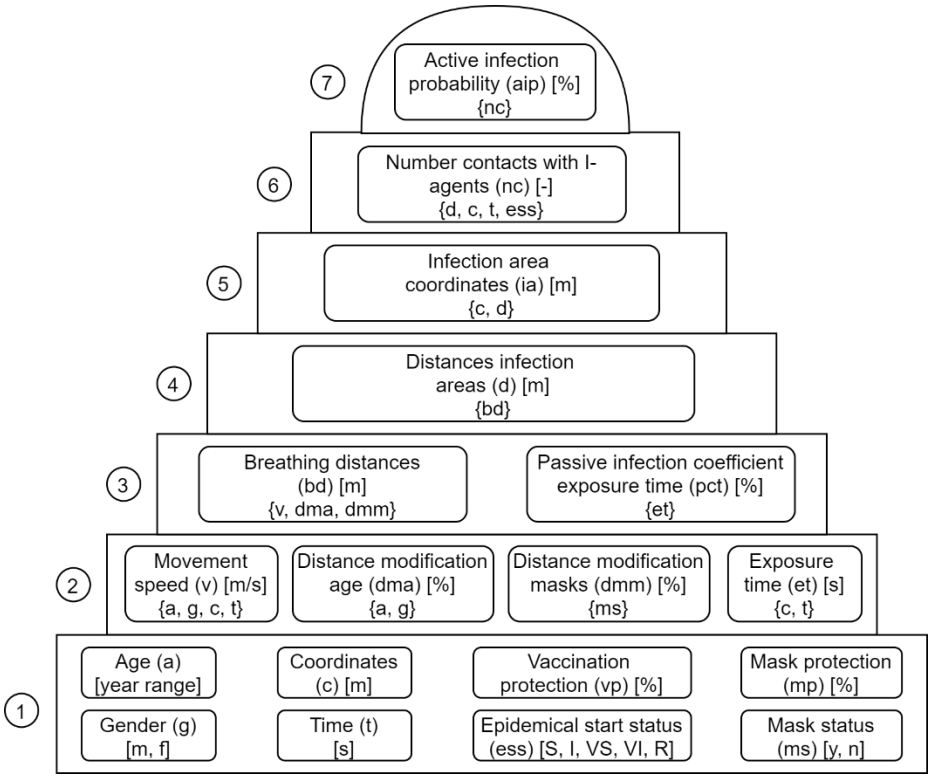

Figure 10 - Agent attributes hierarchy

Figure 10 shows the agent attributes hierarchy. The lowest level contains the attributes age and gender (male or female), forming the agent profile for the movement simulation. For the movement simulation, time-dependent (x- and y-) coordinates are assigned to the agents. The time difference between two simulation steps is 0.25 s. In addition, each agent is specified by his epidemical start status (susceptible, infectious, vaccinated and susceptible, vaccinated and infected, or recovered) and his (face) mask status (yes = wearing a face mask, no = not wearing a facemask). The efficiency of vaccinations and face masks is represented by the vaccination protection, respectively the mask protection percentage.

On the second level, the agent profiles (age and gender) are evaluated for the movement speed and a modification factor for the distances of the infection areas around I-agents (distance modification age). The usage of face masks (mask status) influences the infection areas through the distance modification masks factor. Within the movement simulation, there is a logical mutual interaction of the time-dependent coordinates and the movement speed. Two sets of each two coordinates with known time differences can, as well, be used to calculate the movement speed, as start-coordinates and movement speed (vector) can be used to calculate end coordinates. Time and coordinates also indicate when an agent enters and leaves the simulated area – the connected time difference results in the exposure time.

The third level combines the movement speed and both distance modification factors to calculate the distances up to which the dose of infectious droplets in the exhaled breathing air of I-agents can be contagious (breathing distances). In this calculation, additional constant factors are used to modify the distances together with a normally distributed factor (mean value = 1) to account for the natural diversity of breathing strength. The individual exposure time influences the passive infection probability by the passive infection coefficient exposure time. More about the passive infection mechanism can be found in Figure 11 and below.

Levels four to seven contain the attributes on the final steps to the active infection probability. Based on the third level breathing distances, the distances of the infection areas are calculated as edge points of a kite and later of an octagon. More precisely, distances for three kites are calculated to represent the three different considered contact types: close, intermediate, and casual. Afterwards, these distances are added to the agents' locations (coordinates) to receive the coordinates of the infection areas. Comparing the locations of the S-agents with the infection areas around the I-agents, the number of contacts can be counted as crossings of the infection areas, dependent on the vaccination status of the I-agents. Lastly, the resulting individual active infection probability is estimated with respect to the number of contacts.

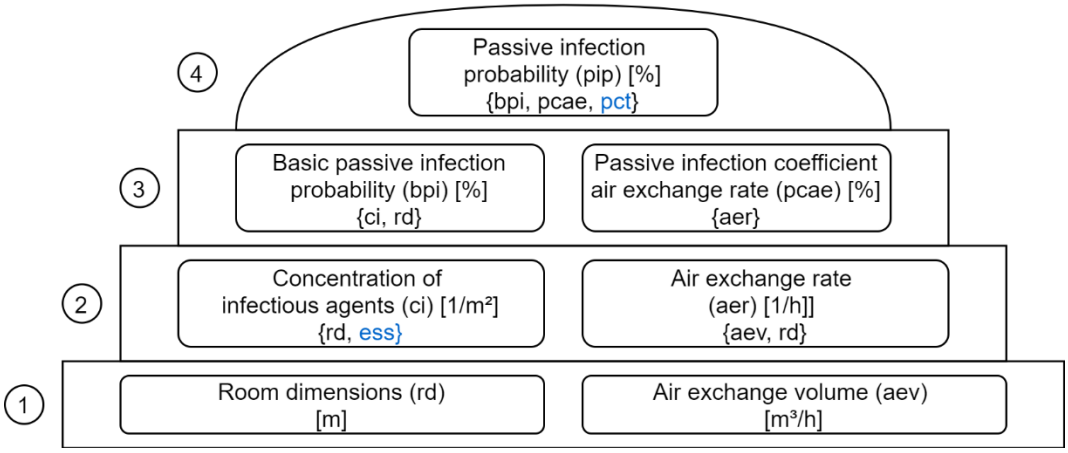

Figure 11 - Environment attributes hierarchy

The environment attributes presented in Figure 11 contain attributes not directly linked to individual agents, but to the agent population as a whole and their surroundings. Connections to agent attributes are highlighted in blue font colour. On the first level, these are the room dimensions (lengths, widths, and heights) and the air exchange volume caused by natural ventilation and ventilation systems. The next higher level combines the room dimensions with the epidemical start status of the agents and the air exchange volume to the concentration of I-agents and the air exchange rate. The concentration of I-agents describes how many I-agents are located per square meter room. The air exchange rate describes how many times the air within a room is replaced with fresh air every hour.

On the higher levels, the passive infection probability is calculated. A basic passive infection probability is defined based on the concentration of I-agents and the room heights. This value is adjusted by two passive infection coefficients, one for the air exchange rate and one for the exposure time of each agent.

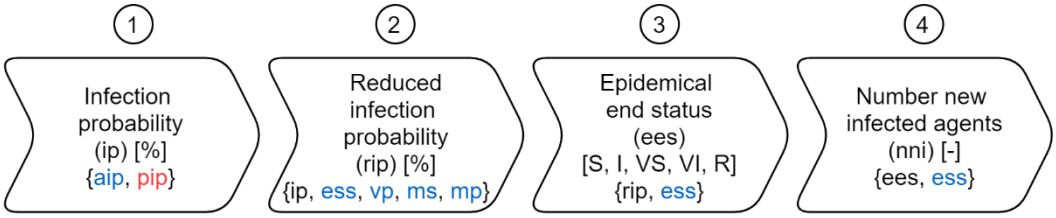

Figure 12 - Concluding attributes hierarchy

As shown in Figure 12, the active and passive infection probabilities are added in the general infection probability. If an S-agent is vaccinated and/or wearing a face mask, this probability is reduced by additional factors. Afterwards, the individual infection probabilities are

concluded, resulting in an epidemical end status which may differ from the epidemical start status. Finally, the number of new infected agents is the difference between infected agents at the beginning and the end of the simulation.

6.1.3 Process overview and scheduling

While the previous chapter focused on the attributes and their definitions, this chapter describes the underlying processes to obtain the attributes. Here, two different processes are defined. The movement process is directly connected to agent movement, considering and resulting in the initial (low level) attributes. Then, the evaluation process analyses the obtained movement information and evaluates it, resulting in the high level and concluding attributes.

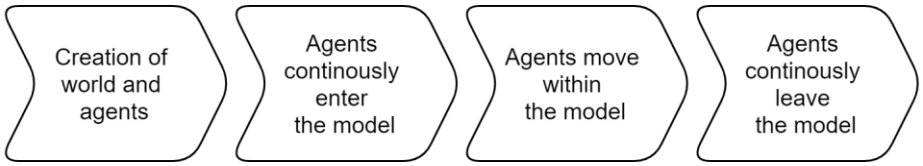

Figure 13 - Movement process

The simplified movement process is shown in Figure 13 and starts with the creating the world and the agents considering all their initial attributes. Afterwards, the simulation starts, and agents continuously enter the room using the entrance doors. The agents move within the model according to the predefined movement rules. In the standard setting, the agents directly move towards the closest exit door and continuously leave the model. The simulation ends when the last agent leaves.

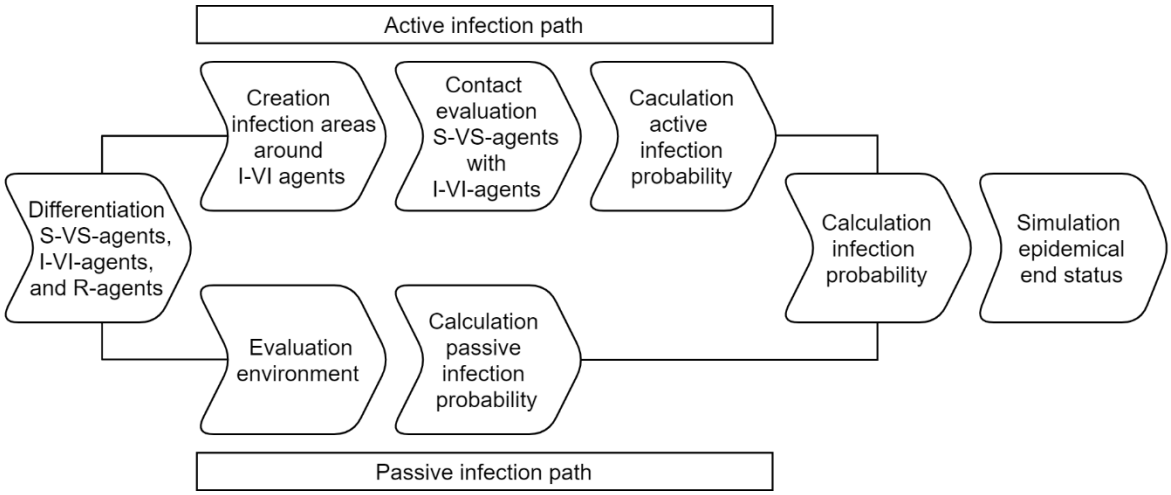

Figure 14 - Evaluation process

For the evaluation process presented in Figure 14, the agents are grouped depending on their epidemical status. The first group consists of S-agents (vaccinated and unvaccinated) and the second of their potential infectors. The R-agents in the third group are non-infectious and immune to the disease and, therefore, not further considered. Next, the process is split into two paths, the active (upper) and the passive (lower) infection path.

On the active infection path, time-dependent infection areas are formed around the I-agents. For each time step, every I-agent is surrounded by three octagons for close, intermediate, and casual contact. If an S-agent crosses the casual contact infection area, it is checked whether he also crosses the intermediate contact infection area and, if that is the case,

whether he crosses the close contact infection area. For each crossing, a value is added to the number of contacts the S-agent gathers during the simulation. The value depends on the type of infection area crossed and the vaccination status of the potential infector. At the end of this path, the active infection probability is estimated based on the number of contacts.

On the passive infection path, the environmental attributes are evaluated together with the number of I-agents, resulting in passive infection coefficients. In combination with the basic passive infection probability and its thresholds, the general passive infection probability is calculated.

The active and passive infection probabilities are summed up to the general infection probability that is reduced for vaccinated and/or mask-wearing S-agents. The epidemical end status as the final outcome is simulated using the (reduced) infection probability. The end status can be compared to the start status, resulting in the number of new infected agents and similar values used to compare the results of different simulations. All processes are repeated 20 times to obtain result distributions rather than single values.

6.2 Design concepts

The first design concept section is called emergence and is used to describe the triggers of specific agent behaviours by dividing them into the categories pre-defined (directly coerced by the modeller) and simulated behaviour (developing during the simulation). In the model created in this report, the number of agents and the location of the doors as entry and exit points are pre-defined. Everything in between, e. g., the route selection and interaction of the agents, is simulated.

Next, the adoption section addresses the agents' preventive (adaptive) behaviour selected to reduce the personal infection risk. The adapted behaviour can be chosen explicitly or based on subconscious decisions. Modelled explicitly-chosen preventive behaviours are wearing a face mask and/or be vaccinated.

The third section named fitness describes to what degree agent behaviour is goal-seeking and how much this behaviour influences the simulation outcome. That again refers to the implemented preventive behaviour. The efficiency of these measures depends on specific diseases and vaccines but is, generally, expected to be significant. Goal-seeking behaviour implies that the agents are informed about the current situation, to reflect this information in their behaviour. In the model, the agents know there is a disease and may be or become infected. Contrarily, the agents do not know their epidemical status, e. g. due to being asymptomatic or prone to optimistic biases. Therefore, the adaption of preventive behaviour is not modelled based on the epidemical status.

The next section (prediction) contains information about the agents' expectations (predictions) of which consequences their decisions might have. Predictions are, typically, driven by (own) experience and personal attitude. The modelled low mask-wearing and vaccination percentages relate to risks perceived as low among the general public.

The interaction section highlights the influence the agents have on each other. In the standard model, there is no intentional interaction among the agents. They only try to reach the exit door and may randomly come into contact with other agents on their way. Other Agent interactions are added in the scenarios (chapter 6.5).

The agents' perception of their environment is the topic of the next section (sensing). While the epidemical status is not perceivable, the agents can generally guess the profiles of other agents and their mask-wearing status. No influence of this perception is modelled.

In contrast, the influence of distributed attributes (stochasticity section) is a key aspect of the model. Most input values, incl. agent profiles, movement speed, and epidemical, mask-wearing and vaccination statuses, are assigned following stochastic distributions and percentages. Considering this variability is especially important because of missing exact values and the natural variance of these attributes among the population. Furthermore, distributed input values result in distributed output values creating a more realistic image of the real world that is also prone to uncertainty.

The agent interaction within social groups is described in the collectives section. The standard model does not contain social groups. As for the general agent interaction, that changes in the additional scenarios (chapter 6.5).

The observation section addresses the validation and perspective of the model's results. As mentioned in the methodology (chapter 3), the model is initially validated through extreme cases and a plausibility check of the overall results. A complex validation is not included in this report. The results are observed from a theoretical-omniscient perspective, meaning that the epidemical end status is assigned to agents as an inevitable prediction, even though a potential infection might not be detectable directly after the infectious contact.

6.3 Details

6.3.1 Initialisation

The basis of the ABM is the creation of the agents. In this case, the agents are first described by their profiles combining age and gender. The agent profiles, their distribution ("Agents" column), and the connected attributes movement speed, shoulder width, and height are presented in Table 2. The movement speed and shoulder width are modelled as log-normal distributed ranges with the mean value μ and standard deviation σ.

Profile [a, m/f]	Agents [%]	Movement speed (μ, σ) [m/s]	Shoulder width (μ, σ) [cm]	Height [cm]
2-5 m	2.60	0.30-1.29 (0.49, 0.30)	10.00-20.75 (14.75, 3.00)	100
2-5 f	2.60	0.30-1.29 (0.49, 0.30)	10.00-20.75 (14.75, 3.00)	100
6-9 m	2.60	0.50-1.30 (1.00, 0.30)	12.00-24.50 (18.50, 3.00)	130
6-9 f	2.60	0.50-1.30 (1.00, 0.30)	12.00-24.50 (18.50, 3.00)	130
10-19 m	8.50	0.50-1.70 (1.30, 0.20)	32.00-48.00 (40.00, 4.00)	140
10-19 f	8.50	0.50-1.70 (1.30, 0.20)	28.00-44.00 (36.00, 4.00)	140
20-29 m	6.00	0.50-2.50 (1.40, 0.20)	32.00-48.00 (40.00, 4.00)	182
20-29 f	6.00	0.50-2.50 (1.40, 0.20)	28.00-44.00 (36.00, 4.00)	162
30-39 m	6.30	0.50-2.50 (1.50, 0.20)	32.00-48.00 (40.00, 4.00)	182
30-39 f	6.30	0.50-2.30 (1.40, 0.20)	28.00-44.00 (36.00, 4.00)	162
40-49 m	6.30	0.50-2.50 (1.50, 0.20)	32.00-48.00 (40.00, 4.00)	182
40-49 f	6.30	0.50-2.10 (1.40, 0.20)	28.00-44.00 (36.00, 4.00)	162
50-59 m	7.20	0.50-2.10 (1.40, 0.20)	32.00-48.00 (40.00, 4.00)	182
50-59 f	7.20	0.50-2.00 (1.40, 0.20)	28.00-44.00 (36.00, 4.00)	162
60-69 m	6.10	0.50-1.90 (1.40, 0.30)	32.00-48.00 (40.00, 4.00)	182

60-69 f	6.10	0.50-1.80 (1.30, 0.30)	28.00-44.00 (36.00, 4.00)	162
70+ m	4.40	0.50-1.70 (1.30, 0.30)	32.00-48.00 (40.00, 4.00)	178
70+ f	4.40	0.50-1.70 (1.30, 0.30)	28.00-44.00 (36.00, 4.00)	164

Table 2 - Agent profiles and standard distribution, based on (Al-Azawi et al., 2021)

The agent profile and movement speed have a direct influence on many simulation steps and higher-level attributes. The shoulder width only indirectly influences the movement of the agents, setting limits on how close the agents can get to each other. The agent height does not influence other attributes and is only for graphical representations of the model.

The agents are further characterised by their epidemical status and mask (-wearing) status. The assignment of these attributes is according to the percentages in Table 3 and Table 4. Among the epidemical statuses, the vaccination status is assigned independently of whether an agent is infectious. Therefore, in Table 3, the status *vaccinated* (V) groups the vaccinated-susceptible and the vaccinated-infectious-agents. The mask-wearing percentage is the same for all agents, including the recovered ones. The percentage of I-agents is assigned relatively high and the mask-wearing percentage relatively low, to receive evaluable results even though the duration of the simulations is short concerning the available computation power and time.

Epidemical status	Agents [%]
S	45
I	50
R	5
V (VS or VI)	10

Table 3 - Epidemical start status agents

Mask-wearing	Agents [%]
Yes	10
No	90

Table 4 - Mask-wearing percentage

The environment in which the agents move is modelled as five times five meters room (Figure 15) with two entrances (red) and one exit door (green). All doors are one meter wide and located in the middle of a wall. The air exchange rate is set to 2 / h. The agents are placed in waiting areas (orange) leading to the entrance doors in random order. Agent movement within the waiting areas is not considered during the evaluation. When the simulation starts, one agent per second enters the evaluated room (green) through each entrance door and starts moving towards the exit. The waiting areas are modelled so small that there is always at least one agent waiting in front of the entrance door until the last one enters the room. The next steps, beginning with the creation of the infection areas and the contact evaluation, are described as part of the submodels in chapter 6.3.3.

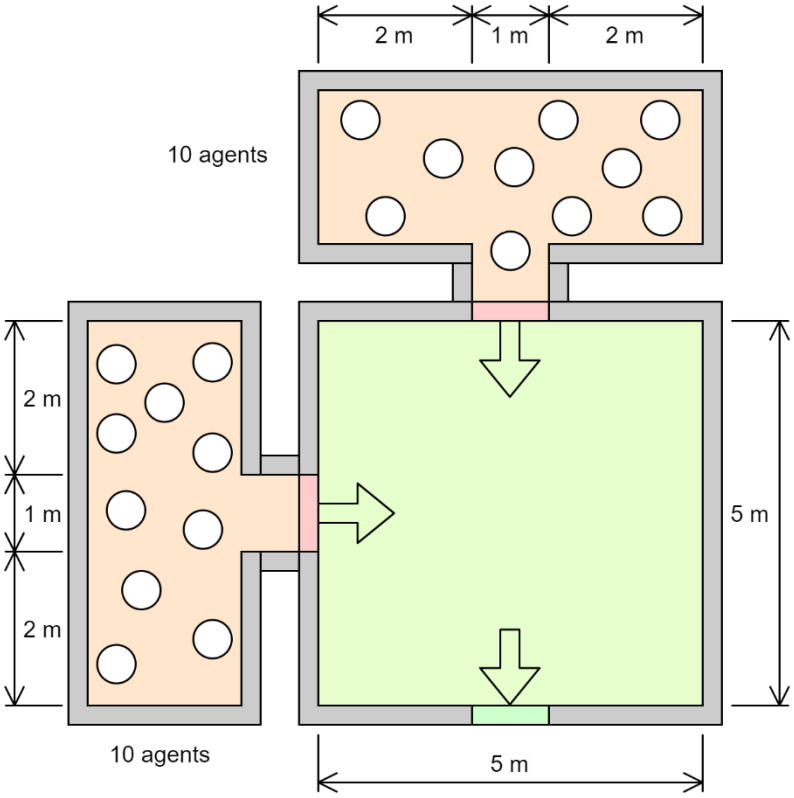

Figure 15 - Sketch small model

6.3.2 Input

The creation of the model is built upon the information obtained through the literature review. It is a representation of the current state of the art in disease modelling (chapter 5.1.2). As the first objective for this project is to develop a model to estimate transmission rates for airborne diseases in general – neither for a specific disease nor a real place, many input variables are to be understood as placeholders to make the model functional. These are to be replaced in case of further specific usage in the future. In an application case, all attributes should be assigned based on the specific disease and place analysed.

The agent profiles, their movement speed, shoulder width, and height in Table 2 were developed during a previous semester project at Aalborg University Esbjerg (Al-Azawi et al., 2021) about the Fehmarnbelt tunnel. Therefore, the values, especially the demographics, refer to the population in Germany and Denmark.

6.3.3 Submodels

Besides the separation in the two process types, movement and evaluation, described in chapter 6.1.3, the model creation can be divided into six self-contained work packages: the room creation (CAD-software), the movement simulation (evacuation simulation software), and four programming work packages. The programming work is written in R and consists of the four scripts: agent profile parameters, interpolation and infection areas, crossings infection areas, and probability calculations. The repetitions of the simulations are automated using python code. R is used for the programmed part of the result evaluation.

The first work package, the drawing of the room resulting in a three-dimensional version of Figure 15, only requires basic knowledge in how to use this kind of software and is, therefore, not further described.

For the movement simulation, a software called Pathfinder is used. Other software is also usable if agent profiles can be assigned and are connected to the simulation results that must include the coordinates for each agent at every time step. In Pathfinder, the needed output must be selected before running the simulation. Other than that, the preparations include: generating the evaluated room plus the waiting areas and adding the agents and doors. After the simulation, a combination of agent (id) and agent profile is printed in the file *original file name_occupant_params.csv.* The other results, including coordinates and movement speed, are in the file *original file name_occupants_detailed.csv*.

6.3.3.1 Script 0 – Agent profile parameters

The *original file name_occupant_params.csv* file is the basis for the further agent profile modelling in programming Script 0 (appendix 12.3.1). Depending on the age of the agents, (infection) distance modification factors are assigned as shown in Table 5. Additionally, each agent receives his epidemical start status and mask (-wearing) status as selected by random sampling based on the percentages from Table 3 and Table 4. All agent attributes are combined in one table and exported as *Agent parameters.txt*.

Age [a]	Distance modification [%]
2-5	25
6-9	50
10-49	100
50-69	75
70+	50

Table 5 - Distance modification factors age

6.3.3.2 Script 1 – Interpolation and infection areas

Next, *Agent parameters.txt* and *original file name_occupants_detailed.csv* are imported in Script 1 (appendix 12.3.2) to interpolate the coordinates of the agents and form the infection areas. The movement simulation, originally, uses time steps of one second in which the agents can move up to 2.5 meters. To reduce the distance travelled between two simulation steps, time, coordinates, and movement speed are linearly interpolated, resulting in time steps of one-quarter of a second. For I- and VI-agents, the distances for the infection areas are calculated based on a linear correlation of breathing distances and movement speed, as shown in Figure 16 and the equations below. The front and back breathing distances are modelled as linear functions dependent on the movement speed using the extreme cases of no movement and maximum speed. If the agent is not moving, his exhaled air reaches a maximum distance (0.8 m) in front of him and is not blown to his backside, assuming only negligible wind speeds (indoor). If the agent is moving with maximum speed, the air stream reduces the front breathing distance to 50 % while reaching the maximum back breathing distance as 350 % of the maximal front breathing distance. To account for the natural variability, the calculated breathing distances are, afterwards, multiplied with a normally distributed gamma parameter ($\mu = 1$, $\sigma = 0.05$).

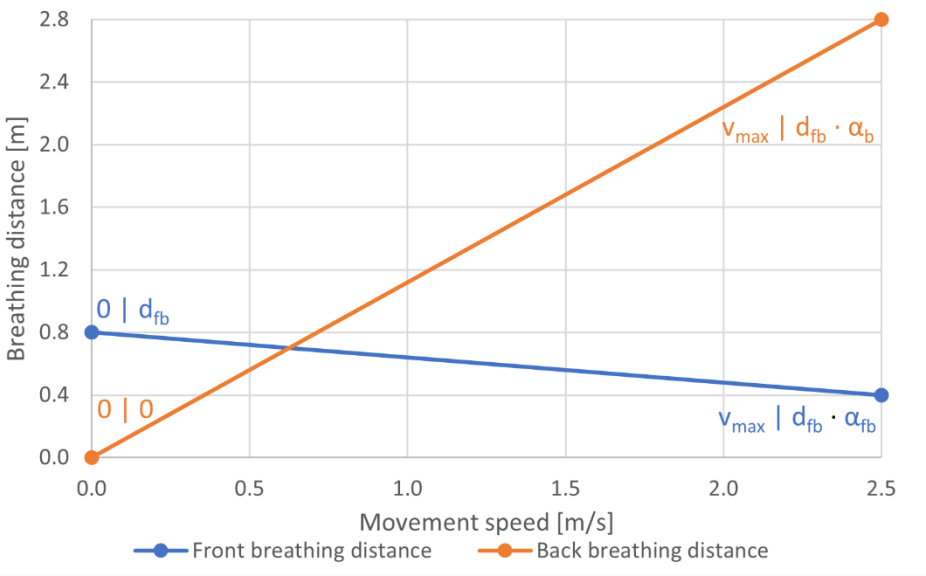

Figure 16 - Correlation movement speed - breathing distances

d_f $= Front\ breathing\ distance\ [m]$, $d_{fb} = Maximal\ front\ breathing\ distance [m]$
$\alpha_{fb} = Minimal\ front\ breathing\ factor\ [\%]$, d_b $= Back\ breathing\ distance\ [m]$,
v $= Movement\ speed\ [m/s]$

$$d_f(v) = d_{fb} \cdot \alpha_{fb} - \frac{d_{fb} \cdot \alpha_{fb} - d_{fb}}{v_{max} - 0} \cdot (v_{max} - v)$$

$$d_f(v) = 0.8 \cdot 0.5 - \frac{0.8 \cdot 0.5 - 0.8}{2.5 - 0} \cdot (2.5 - v) = 0.4 + 0.16 \cdot (2.5 - v)$$

$$d_b(v) = \frac{d_{fb} \cdot \alpha_b - 0}{v_{max} - 0} \cdot (v - 0) + 0 = \frac{0.8 \cdot 3.5 - 0}{2.5 - 0} \cdot (v - 0) + 0 = 1.12 \cdot v$$

For agents wearing a face mask, the front and back breathing distances are reduced to 20 % of the original value. Completing the distances for the initial infection area kite, the breathing distances on both sides of the agent are set to 0.5 m. All these distances are modified by the distance modification factor age (Table 5).

To receive the coordinates of the infection area, the calculated distances and the angle between the movement direction and the x-axis are used. The angle is defined by the next equation. In addition to this angle, the distances can have different mathematical signs (+/-) dependent on the movement direction. An example of this influence is shown in Figure 17.

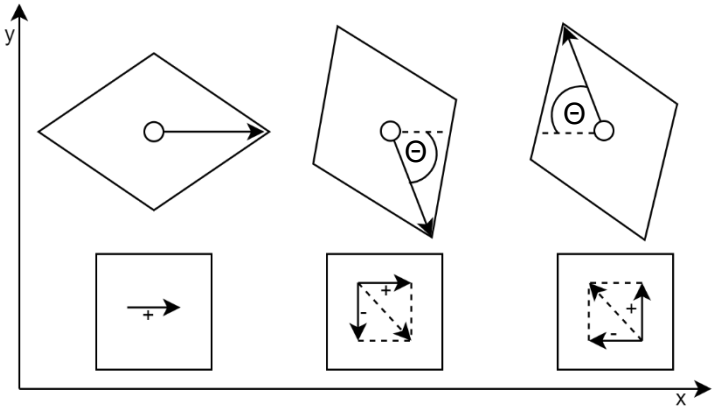

Figure 17 - Mathematical signs infection distances

$$\Theta_i[rad] = arctan\left(\frac{y(t_i + \Delta t) - y(t_i)}{x(t_i + \Delta t) - x(t_i)}\right)$$

In Figure 17, the arrows in the three depicted infection areas represent the movement direction of the agent and, therefore, the front edge point of the area. The arrows and mathematical signs in the squares below indicate how the distances agent centre - front point are added to the agent coordinates to receive the ones for the front point. In the first case (Figure 17 left), the agent moves parallelly in the direction of the x-axis, so the distance on the x-axis is added to the agent's x-coordinate, and the y-coordinate remains the same. In the second case (Figure 17 middle), the agent moves with the angle Θ to the x-axis. The x-part of the distance to the front point is again positive, and the y-part is negative because it points in the opposite direction of the y-axis. The angle Θ is the same in the third case (Figure 17 right), but the movement direction is the opposite (x-part in the opposite direction of the x-axis, y-part in the same direction as the y-axis). There are analogue correlations for the other edge points of the kite. These are modelled using if-loops in the code. All cases are listed in the programming script (# Determination of mathematical sign).

The four additional edge points for the area transformation to an octagon are in the middle of each side of a temporarily expanded kite version (expansion factor 1.2). Figure 18 provides a sketch of the resulting points. The original kite around the agent is black, the expanded version grey, and the octagon blue. In Script 1, the distances from the agent centre are calculated by the equation below. For each infected agent and time step, the equation is used eight times, obtaining the x- and y-distances of all four additional points on the left and the right, in front of and behind the agent.

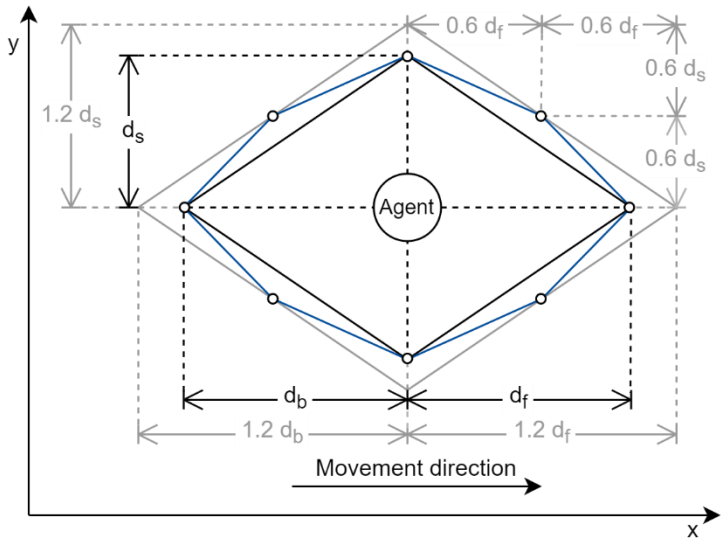

Figure 18 - Infection area - Transformation to octagon

$$d_{M,i,j,k}[m] = 1.2 \cdot d_{i,k} + \frac{1.2 \cdot d_{i,k} - 1.2 \cdot d_{j,k}}{2}$$

$$with \quad i = Orientation\ (front/back), \qquad j = Side\ (left/right), \qquad k = axis\ (x/y)$$

The first-created octagon-shaped infection area represents the close contact (red). Afterwards, two more areas are modelled by enlarging the close contact distances with factors of 1.5 for intermediate (green) and 2.0 for casual contact (blue). The resulting infection areas around I-agents are presented in the colours in brackets in Figure 19. The output of Script 1 is called *Revised dataset.txt* and contains the agent parameters id, profile, epidemical start status, location, and the edge points of the infection areas for every time step of the simulation.

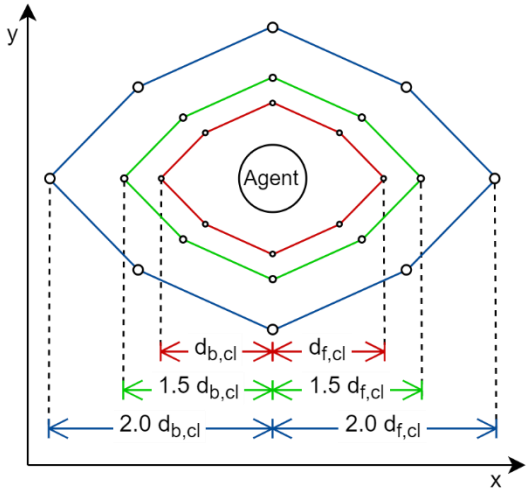

Figure 19 - Infection area types

6.3.3.3 Script 2 – Crossings infection areas

Based on the results of Script 1, Script 2 (appendix 12.3.3) analyses how many times each S-agent crosses infection areas. The underlying procedure is a contact counting approach. Whenever an S-agent crosses an infection area, his number of contacts is increased by a certain value. The value depends on the type of infection area crossed (close, intermediate, or casual contact) and the vaccination status of the I-agent. A flow diagram of this analysis representing the applied programming code is shown in Figure 20. In the figure, the compared agents are placed in oval boxes, attributes in angular rectangles, and calculations in rounded rectangles. The order and connection of the steps are indicated by arrows. Solid line arrows have the highest priority and are relevant unless the condition on dashed line arrows is fulfilled.

The approach starts with the first S-agent (s = 1) and his location at the beginning of the simulation ($t_s = 0$) compared with the first casual infection area ($c_3 = 1$) of the first I-agent (i = 1) at the time t_i. The beginning of the simulation is defined as the moment the first agent enters the evaluated room. If the S-agent is within the infection area, it is checked whether the infection area is connected to the same simulation time or one simulation step before ($0 \leq t_s - t_i \leq \Delta t$). If that is also the case, the number of contacts for agent s is increased by 0.25 times the factor vs that depends on the vaccination status of agent i (vs = 0.5 if agent i is vaccinated, otherwise vs = 1.0). The analysis continues with the intermediate contact infection area as the next smaller infection area type. If at least one condition is not met, the next casual contact area ($c_3 = c_3 + 1$) is compared with the S-agent's location at t_s. This is repeated until all casual contact infection areas of agent i are compared ($c_3 > c_{3,max}$). Then, the same is done for the next I-agent (i = i + 1). When all I-agents are considered ($i > i_{max}$), the location of agent s is replaced with the one at the next time step ($t_s = t_s + \Delta t$). After the last time step is set-in, the contact counting for agent s is finished, and the contacts of the next S-agent are evaluated (s = s +1). The contact counting procedure ends with the evaluation of the last S-agent ($s > s_{max}$). The procedure is analogous for the intermediate and close contact infection areas. In the case of intermediate contact, the contact value is additionally increased by 0.25 vs, and, in the case of close contact, increased a third time by 0.50 vs.

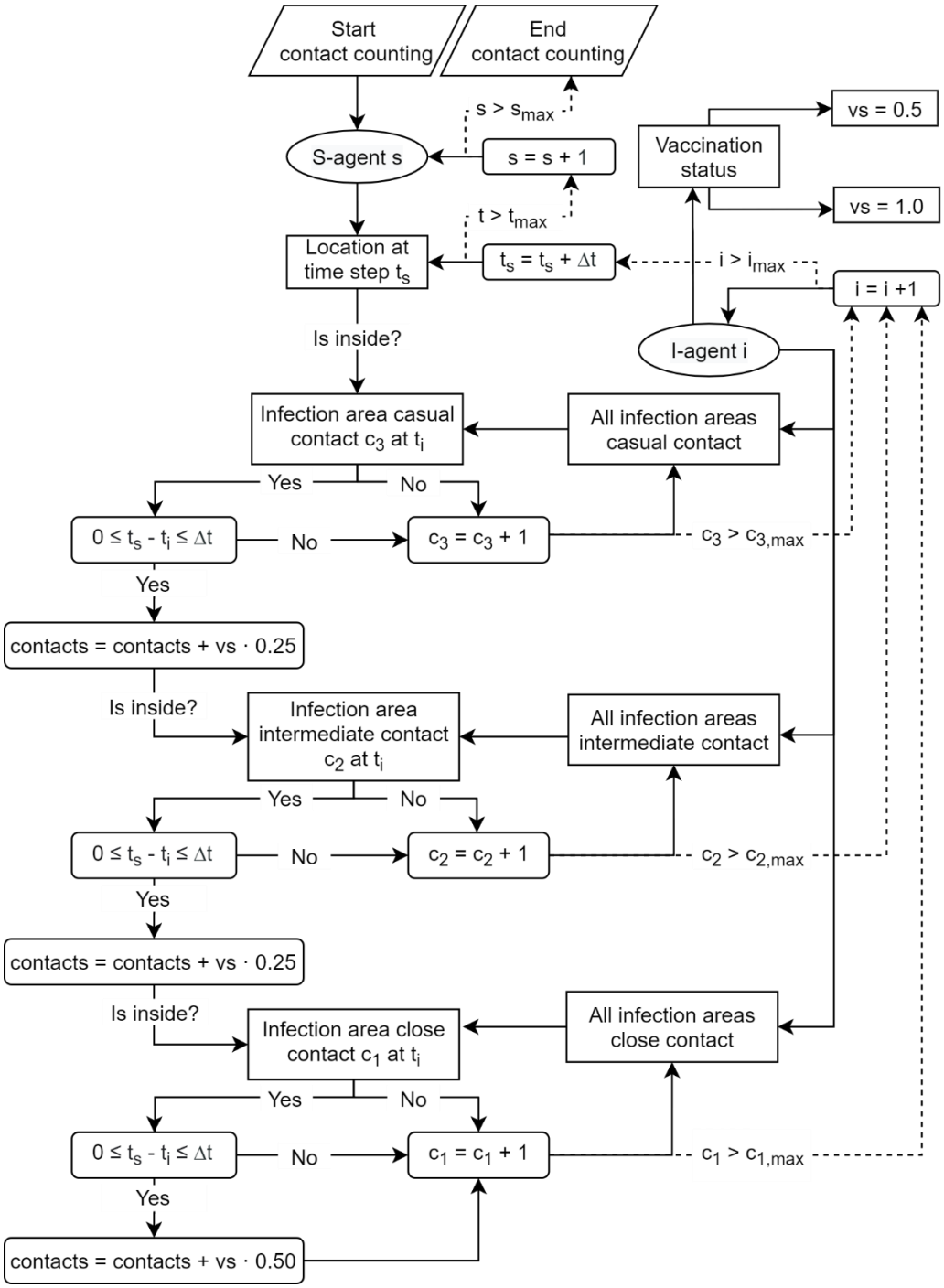

Figure 20 - Flow diagram contact counting approach

To evaluate whether the agent is inside an infection area, eight triangle-sub-areas are defined by the agent centre (as a point) and each two edge points of the infection area. As shown in Figure 21, the resulting sub-areas look like the infection area is cut into slices by cutting from each edge point towards the agent centre. That is only the case if the agent centre is inside the infection area. This visual impression is expressed in numbers by adding the sub-areas – if the agent centre is inside, the sum of the sub-areas equals the total infection area.

In general, to calculate the area of polygons, the x- and y-coordinates of all edge points is cross-multiplied, resulting in the support variables S_1 and S_2. Afterwards, the polygon area equals half of the difference between S_1 and S_2. The equations are:

$$S_1 = \sum y_i \cdot x_{i+1} \qquad S_2 = \sum x_i \cdot y_{i+1}$$

$$A = \frac{S_1 - S_2}{2}$$

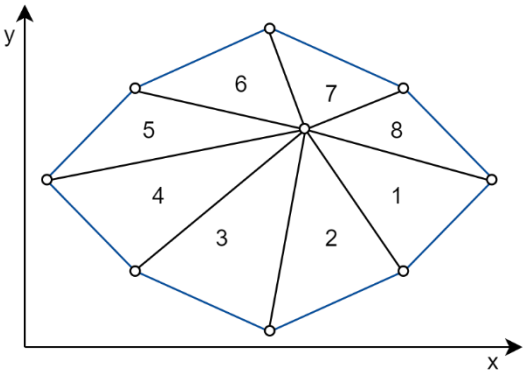

Figure 21 - Sub-areas point in octagon

The output of Script 2 is two tables, one with detailed information (*All geometric results.txt*) and one only containing the main results (*Basic geometric results.txt*) as needed as input for Script 3. The main results table lists the agent ids, agent profiles, mask-wearing statuses, epidemical start statuses, start times (the agents enter the room), end times (the agents leave the room), the resulting exposure times (end time – start time), and the contact values. *All geometric results.txt* adds all other previously assigned or evaluated coordinates.

6.3.3.4 Script 3 – Probability calculations

R programming Script 3 (appendix 12.3.4) concludes the information achieved in the previous scripts resulting in individual infection probabilities and their outcome (epidemical end status) for each agent. Before the active and passive infection probabilities are added up, they are evaluated separately.

The active infection probability is assigned based on the ranges of the number of contacts stated in Table 6. If the evaluated S-agent had no contact with I-agents ($n_c = 0$), he does not become actively infected. However, due to value restrictions in the used programming code, the assigned value must be greater than zero and is, therefore, selected negligible small (0.0000001).

Number of contacts (n_c) [-]	Active infection probability [%]
$n_c = 0$	~ 0
$n_c \leq 3$	25
$3 < n_c \leq 8$	50
$8 < n_c \leq 15$	75
$n_c > 15$	95

Table 6 - Active infection probability

The final passive infection probability is formed by a base passive infection probability that is adjusted by coefficients considering the air exchange rate of the room (Table 7 left) and the exposure time of each agent (Table 7 right). The base passive infection probability considers the concentration of infected agents per square meter and is limited to values between 5 % and 30 %. It is calculated using the following equation. Then, the base passive infection probability and both coefficients are multiplied, resulting in the final passive infection probability.

$$c_{I-A} = Concentration\ of\ infected\ agents, \quad n_{I-A} = Number\ of\ infected\ agents$$

$$A_{room} = Area\ of\ the\ evaluated\ room, \quad h_{room} = Height\ of\ the\ evaluated\ room$$

$$c_{I-A} = \frac{n_{I-A}}{A_{room}} \qquad p_{passive,base} = \frac{c_{I-A}}{2 \cdot h_{room}}$$

Air exchange rate (r_{ae}) [1/h]	Coefficient air exchange [%]
$r_{ae} < 1$	100
$1 \leq r_{ae} \leq 3$	80
$3 < r_{ae} \leq 6$	60
$r_{ae} > 6$	50

Exposure time / 60 (t_{exp}) [min]	Coefficient exposure time [%]
$t_{exp} < 1$	50
$1 \leq t_{exp} \leq 3$	60
$3 < t_{exp} \leq 6$	80
$t_{exp} > 6$	100

Table 7 - Coefficients passive infection probability

The active and passive infection probabilities are summed to the general infection probability limited to values between 0 and 99 %. The resulting probability is reduced by a factor of 20 % for agents wearing a face mask and a factor of 50 % for vaccinated agents, considering a mask filter protection percentage of 80 % and a vaccination protection percentage of 50 %.

Estimating the epidemical end status, a random value of either zero or one is generated. If the value is zero, the agent does not become infected. Therefore, the probability of zero is one minus the (reduced) infection probability. Analogous, a value of one is generated with the (reduced) infection probability and means that the agent becomes infected.

All infection probabilities, the epidemical end status, and feedback on whether an agent's epidemical status changed during the simulation are added to the output of Script 2, resulting in *Final results.txt*. This table is the basis for the following result distribution evaluation of repeated simulations and comparing different models and scenarios.

6.3.4 Automation of simulation repetitions

Since many attributes are assigned based on random outcomes of distributions (e. g. movement speed and epidemical start status), the simulation outcomes may differ with every execution. Therefore, each model and scenario is simulated 20 times to achieve more elaborated result distributions. That can be done manually, but since an automated procedure is less time-consuming, this option is preferred. Appendix 12.3.5 contains the used script written in Python programming language. Regardless of whether the repetitions are conducted manually or using the code, the steps are the same and as indicated by Figure 22.

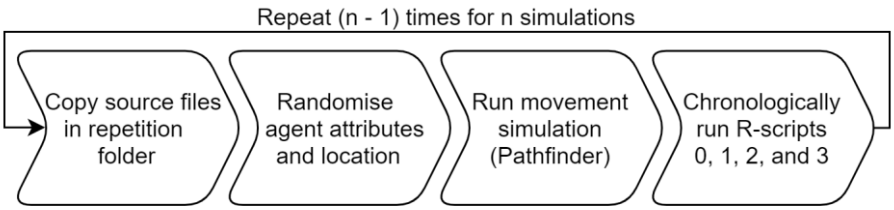

Figure 22 - Simulation repetition steps

First, the source files (Pathfinder model and R-scripts) are copied in a separate folder for every repetition. Afterwards, the agent attributes and location (order of entry to the evaluated room) are randomised using functions implemented in the movement simulation software that then executes the simulation. Next, the output of the movement simulation is evaluated by the R-scripts described in the chapters 6.3.3.1 to 6.3.3.4. The scripts must be executed in chronological order, always waiting for the previous one to be finished, because its output is needed for the following script. Lastly, all steps are repeated (n – 1) times resulting in n conducted simulations.

6.3.5 General result evaluation

Similar to the automation of the repetitions, the first part of the result evaluation can be supported by programming code avoiding a manual read-through of the *Final results.txt* files. The applied script is in the appendix (chapter 12.3.6). The structure of the code is based on a loop that opens the results of all repetitions step by step from the first to the last. For each repetition, the simulation time, minimum, mean, and maximum of the exposure times, the sum of the total contacts, the number of S-agents without contacts, the maximum number of contacts per agent, the maximum passive infection probability, and the number of the new infected agents are combined. The script exports two tables. The first one is named *Summary of outcome distribution.txt* and only contains the before-listed attributes, while the second, *Results of all repetitions.txt*, adds all *Final results.txt* files in one table.

6.4 Conversion to public space model

Based on the initial small model described before, a larger public space model is developed. The main changes are connected to the evaluated room as the agents' environment – unless otherwise stated below, the approach and most input values remain unchanged. In the new model, the room area is quadratic with a side length of 20 m, and the number of agents increases to 200. During the simulation, the agents enter the room through three doors, all placed in one-third of the room (the upper third in Figure 23), and move towards the three exit doors in the middle of the fourth wall. First, the room is empty, but in the second step, eight obstacles are added, as highlighted by the red boxes in the figure. The agents cannot pass the obstacles and must move around on their way to the exit doors. All scenarios introduced in chapter 6.5 use the public space model with obstacles.

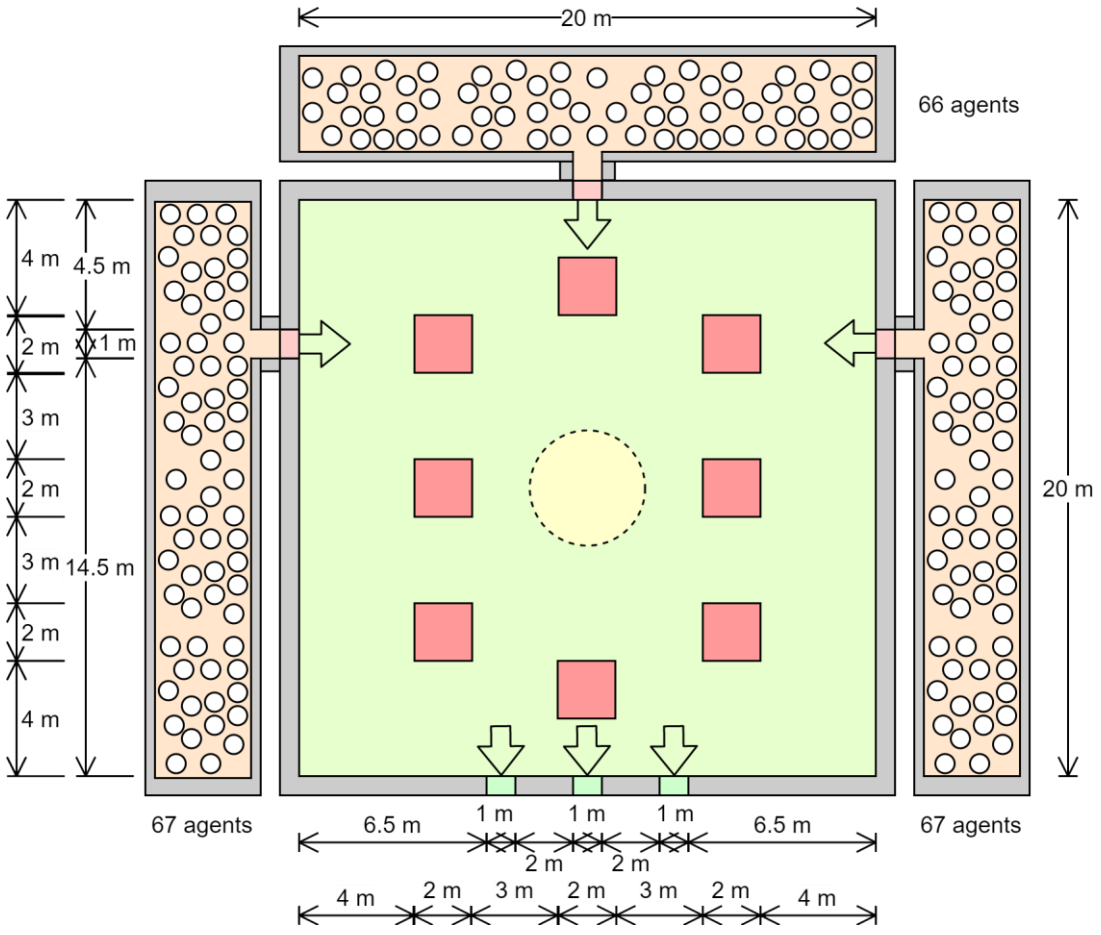

Figure 23 - Sketch public space model

In a real environment, the obstacles could, e. g., be concrete columns or exhibition stands. In the figure, the inner measuring chains on the left and the bottom show the position of the doors, while the outer ones locate the obstacles.

6.5 Scenarios

In the original model, the agents directly move towards the exit and no other behaviour is considered. In the real world, that is not always the case and people may conduct several other actions in public space. Therefore, the first scenario adds a waiting behaviour which could, e. g. represent a queue in front of a market stand. The modelled waiting area is circular with a 2 m radius around the centre of the room, a yellow circle in Figure 23. The agents enter the evaluated area through the entrance doors. In the waiting area, the agents wait between 5 s and 15 s following outcomes of a normal distribution with $\mu = 10$ s and $\sigma = 2$ s. Afterwards, the agents move towards the exit doors and leave the model.

The next scenario analyses the influence of the agent profiles and the connected higher-level attributes on the infection outcome. To do that, only the distribution of age groups changed according to Table 8. For this scenario, the percentage of older agents is increased to a situation as it could be in an elderly home with visiting family members.

Profile [a, m/f]	Agents [%]
2-5 m	1.0
2-5 f	1.0
6-9 m	2.0
6-9 f	2.0
10-19 m	3.0
10-19 f	3.0
20-29 m	4.0
20-29 f	4.0
30-39 m	4.5
30-39 f	4.5
40-49 m	4.5
40-49 f	4.5
50-59 m	6.0
50-59 f	6.0
60-69 m	10.0
60-69 f	10.0
70+ m	15.0
70+ f	15.0

Table 8 - Demographics Scenario 2

Scenario 3 introduces movement groups. In the original model and all other scenarios, every agent moves alone with personal movement speed. In this scenario, 40 % of the agents move alone while 30 % form movement groups of two, 20 % of three, and 10 % of four members. The agents in a group stay close together and wait for other members if the distance in-between increases. Therefore, the groups' movement speed depends on the speed of the slowest member, so that the group generally moves slower than if the agents move alone. For the creation of this scenario, new programming code is added to the simulation repetition file (# Only for movement group scenario). The creation of movement groups follows the randomisation of the agents' location, otherwise the first and the last agent, who enter the evaluated area, could be in the same group, causing the first one to wait at the door all the time.

Every scenario is simulated 20 times with the same input. The results are in the sub-chapters of chapter 7.2.1.

7 Results

7.1 Minimal model

The minimal model is the initial one described in the model representation (chapter 6). According to the percentages stated in Table 2, out of the 20 contained agents, nine are S-agents, ten are I-agents, and one is an R-agent. Out of all agents, two are vaccinated. The distribution of the number of new infected agents for all 20 repetitions of the same setup is presented in Figure 24. In the figure, the blue bars represent, how many times each number of new infected agents (x-axis) is observed. These absolute occurrences are shown on the left y-axis. Dividing these numbers by 20, as the number of repetitions, results in the relative occurrence. Adding up all relative occurrences lower than or equal to a number of new infected agents gives the cumulative relative occurrence for the number. These values form the orange line connected to the right y-axis.

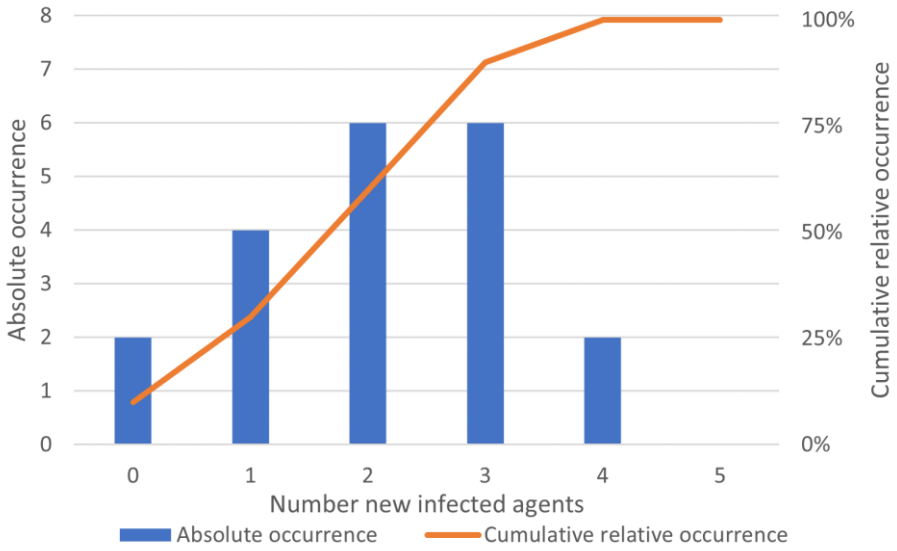

Figure 24 - Number new infected agents - Minimal model

The repetitions result in between zero and four new infected agents during the simulation. Compared to the nine S-agents at the beginning of the simulation, it means that up to 44 % of them become infected. The distribution has a mean value of 2.10, a standard deviation of 1.17, and a median value of 2.00. Further analysis of the simulation outcome shows that the sum of the total number of contacts of all S-agents varies between 3.50 and 19.75 with a mean value of 9.63, a standard deviation of 4.65, and a median of 8.81. The large differences in the number of contacts explain the differences in the numbers of new infected agents. Between none and five (55 %) of the S-agents move through the room without coming in contact with I-agents. Given that an S-agent comes in contact with an I-agent, he has on average between 0.58 and 2.50 contacts (mean value = 1.44).

7.2 Public Space model

In the public space model (described in chapter 6.4), 200 agents are considered, out of which 90 are susceptible, 100 are infectious, and ten are recovered when the simulation starts. The results are divided into the model versions without (Figure 25) and with obstacles (Figure 26). The results of the scenarios from chapter 6.5, using the public space model with obstacles, are described in the sub-chapter 7.2.1.

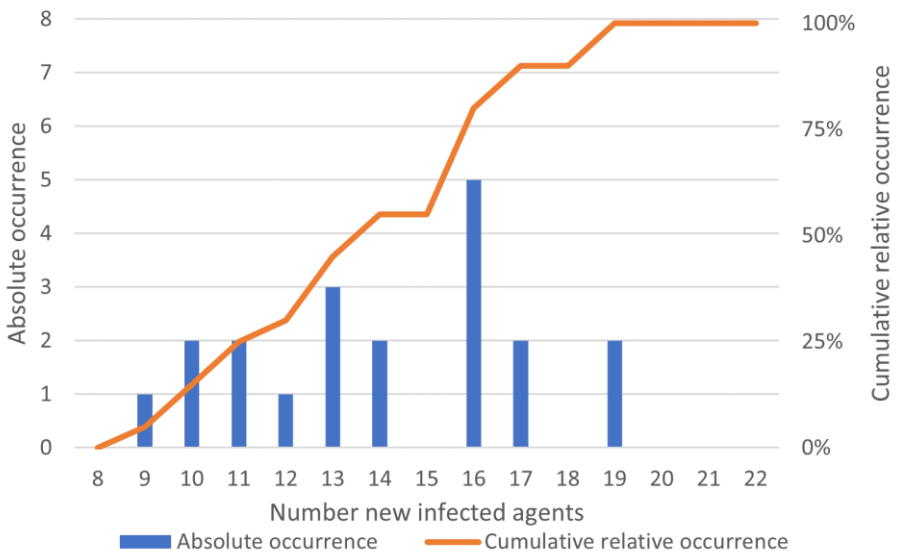

Figure 25 - Number new infected agents - Public space model without obstacles

Without obstacles, nine to 19 S-agents become infected during the simulation, with a mean value of 14.10, a standard deviation of 2.99, and a median value of 14.00. The number of 15 new infected agents is close to the distribution centre but is not an outcome. At the point with the highest slope of the cumulative relative occurrence line (orange) is a dominant mode of 16, occurring twice as often as the median. On average, 47.15 (52 %) of the S-agents leave the room without contact with I-agents. The ones who come in contact with I-agents have up to 4.88 contacts (mean value = 0.70) each, adding up to between 33 and 45 total contacts, with a mean value of 39.05, a standard deviation of 2.92, and a median of 39.00. All agents need between seven and 45 s to reach the exit doors.

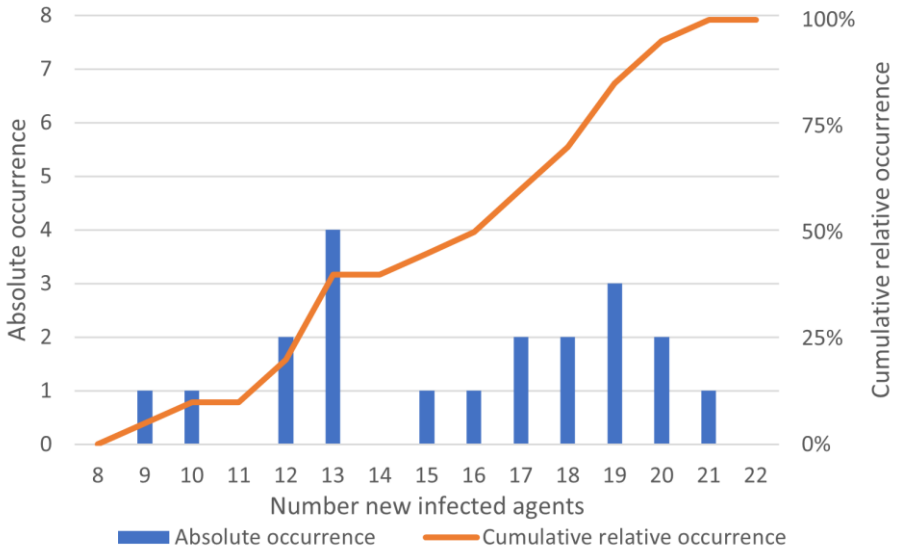

Figure 26 - Number new infected agents - Public space model with obstacles

After adding the obstacles to the room, nine to 21 agents become infected in the simulation process ($\mu = 15.70$, $\sigma = 3.60$, median = 16.50). The complete distribution looks like it consists of two individual distributions, one containing numbers of new infected agents lower than 14 and one of more than 14. The lower one is almost following an exponential function up to 13, while the upper one is bell-shaped with a centre between 18 and 19. That is,

additionally, indicated by the cumulative occurrence line which first rises abruptly and then in a more rounded way. This differentiation is not visible in the other attributes. The total number of contacts of all S-agents varies between 31.63 and 53.13 (μ = 41.34, σ = 6.26, median = 40.88) summing up on average 50.55 agents with 0.82 contacts.

7.2.1 Scenarios

7.2.1.1 Scenario 1 – Including waiting behaviour

As for all other scenarios, the detailed scenario descriptions are in chapter 6.5. In short, Scenario 1 adds a waiting area in the centre of the public space model. All agents move to this area and wait there for a couple of seconds, resulting in a simulated time interval of 117.53 s to 131.53 s. The distribution of the number of new infected agents is in Figure 27.

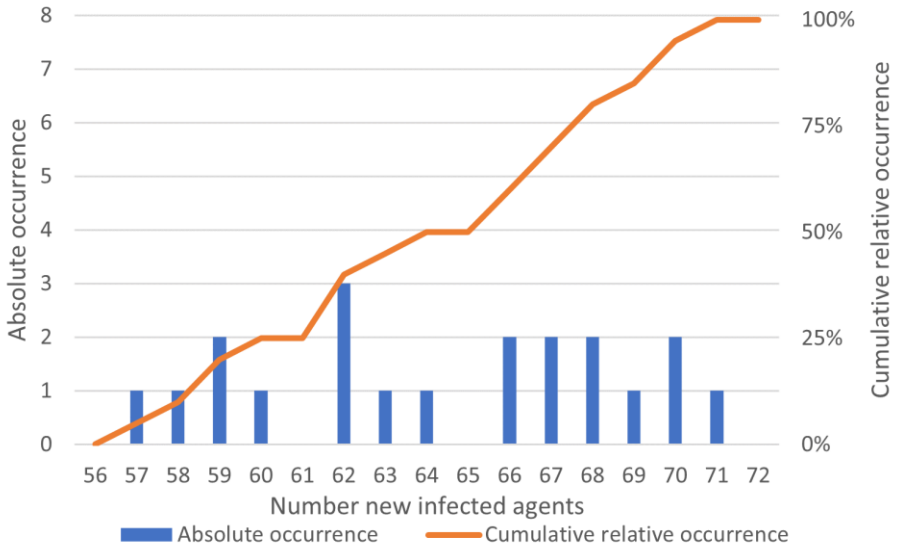

Figure 27 - Number new infected agents - Scenario 1 - Waiting

During the simulation, between 57 and 71 S-agents become infected. As can be seen looking at the absolute occurrence of these values in the figure above, the distribution could be simplified to a uniform type since, overall, the values have a similar occurrence from the first ones to the last ones. That tendency is also visible in the cumulative occurrence which is approximately linear from the beginning to the end. Since in this scenario there is no S-agent without contacts to I-agents, this attribute is exchanged with the minimum number of contacts which is on average 1.20 (min = 0.25, max = 2.25). All S-agents gather in sum up to 1,472.25 total contacts. The number is that high, because each quarter of a second in contact with an I-agent can increase the contact value by up to one. Hence, if an S-agent waits close to an I-agent, he receives an individual contact value of up to 57.50 resulting in a high active infection risk.

7.2.1.2 Scenario 2 – Changing the demographics

For Scenario 2, the standard demographics are changed from the ones listed in Table 2 to those in Table 8 creating an older population. The performance of the new population can be seen in Figure 28.

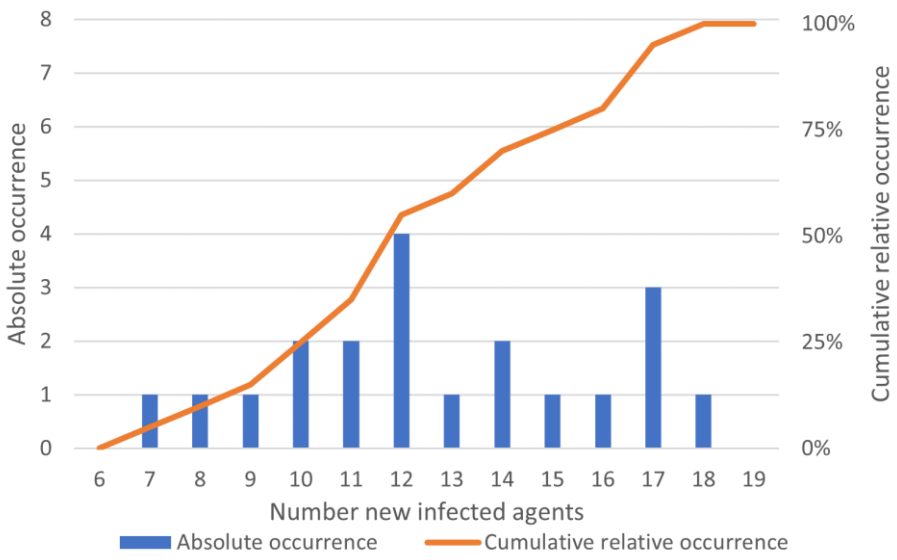

Figure 28 - Number new infected agents - Scenario 2 - Demographics

Figure 28 shows that, using the alternative demographics results on numbers of new infected agents between seven and 18 ($\mu = 12.75$, $\sigma = 3.19$, median = 12.00). All outcomes are almost equally around the median, which, at the same time, is the mode of the distribution. On average, 36.6 of the 90 S-agents (41 %) share 24.49 total contacts, 0.68 contacts per contacted agent.

7.2.1.3 Scenario 3 – Including movement groups

In Scenario 3, the previously alone moving agents are assigned to movement groups according to chapter 6.5. The results of this scenario are presented in Figure 29 below.

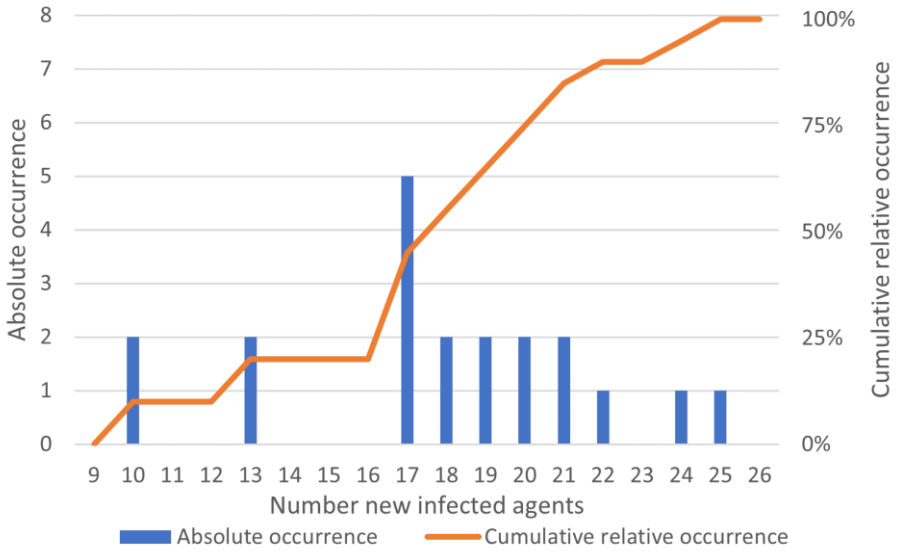

Figure 29 - Number new infected agents - Scenario 3 - Movement groups

Implementing movement groups results in numbers of new infected agents between ten and 25 with a mean of 17.90, a standard deviation of 4.05, and a median of 18. Besides the mode of 17, the occurring numbers up to 21 are equally likely, and the occurring number from 22 up are equally likely again. In between, particularly below the mode, there are several gaps of not occurring numbers where the cumulative occurrence line is horizontal. At the same time, the number of contacts varies between 52.00 and 99.38, with a mean of

75.31, and a high standard deviation of 13.95. A possible reason for this widespread distribution is that the agents stay close together within their movement groups but not necessarily close to other groups. Therefore, the group assignment of the I-agents, done randomly at the beginning of each repetition, influences the outcome. If an S-agent is in a group with (several) I-agents, his number of contacts and the connected infection risk is more likely to be high than if there is no I-agent in his group. If many I-agents are assigned in only a few groups with other I-agents, the overall infection probability is relatively low, because the I-agents in close range are already infected. If, on the other hand, the I-agents are split into many different groups with S-agents, the overall infection probability is comparatively high because then many S-agents come in (close and) continuous contact with the I-agents in their groups.

7.3 Comparison

Based on the results of the single models and scenarios explained before, this chapter compares them and discusses possible reasons for the identified similarities and differences.

For this purpose, the different attributes are converted to comparable scales eliminating the differences in the number of agents. Comparable versions of the main attributes per model, respectively scenario, are shown in Table 9. The connected data sets are in the appendix (chapter 12.4). The first attribute from the left indicates how many of the S-agents become infected. It is the quotient of the number of new infected agents (mean value) and the number of S-agents at the start. The second attribute is the coefficient of variation of the mean value of the number of new infected agents. It is a measurement for the dispersion of the distribution and is calculated by dividing the connected standard deviation by the mean value. Analogous to the first attribute, the third describes the average number of contacts each S-agent has, given he is contacted at all. These values are already mentioned in the individual result evaluation. Lastly, the average percentage of the S-agents without contacts to I-agents is calculated as the quotient of the mean value of the number of S-agents without contacts and the total number of S-agents. The maximum value of each category is highlighted by red font colour, the minimum value by blue font colour.

Model / Scenario	$\frac{\mu_{new\ inf.}}{n_{S-agents}}$	$\frac{\sigma_{new\ inf.}}{\mu_{new\ inf.}}$	$\frac{\mu_{tot.\ contacts}}{n_{S-agents}}$	$\frac{\mu_{no\ contacts}}{n_{S-agents}}$
Minimal	23%	55%	1.44	27%
PS – No obst.	16%	21%	0.70	52%
PS – With obst.	17%	23%	0.82	44%
Scenario 1	72%	7%	14.65	0%
Scenario 2	14%	25%	0.68	59%
Scenario 3	20%	23%	1.15	27%

Table 9 - Attributes comparison models and scenarios

The most noticeable is probably the outcome of Scenario 1, which has the most infections in absolute and relative numbers. While in the other scenarios, around 20 % S-agents become infected, in this scenario, these are 72 % S-agents. It is also the only scenario, in which all of the S-agents are in contact with I-agents. Connected to that, the average number of contacts per S-agent is more than ten times the second-highest value. Considering

the high number of infections, the coefficient of variation for the same is with 7 % very low compared to the others.

The differences between the other scenarios are less extreme. The highest remaining difference is related to the minimal model, which has at 55 % a coefficient of variation more than twice as large as the ones for the other model. Related to that, the average number of contacts per S-agent is also the second-highest after the one for Scenario 1, and the relative quantity of S-agents without contacts is, together with the one for Scenario 3, with 27 % the second-lowest.

The attributes of the other scenarios are comparatively similar. They are ordered by their percentage of new infected agents, as follows: (Scenario 1, minimal model,) Scenario 3, public space model with obstacles ("PS – With obst."), public space model without obstacles ("PS – No obst."), and Scenario 2. This order is graphically presented in Figure 30.

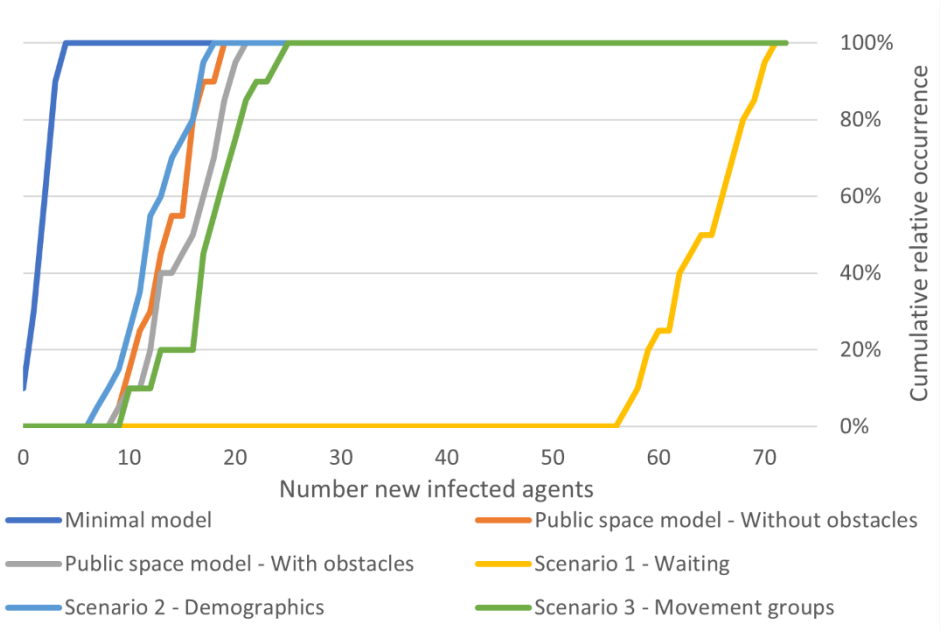

Figure 30 - Number new infected agents - Comparison

The reasons for the high number of new infected agents in Scenario 1 is already explained in chapter 7.2.1.1. If the agents gather at a specific location and wait there, they get in contact with each other, resulting in high infection probabilities. The difference to the other scenarios is, additionally, shown in Figure 31. On the way to the waiting area, the agents keep approximately 1.0 m to 1.5 m distance, but to reach the waiting area, the agents have to move closer to each other. Their distance decreases to, in the centre, between 0.4 m and 0.2 m. When the agents leave the waiting area, their distance slowly increases again. It reaches around 0.6 m to 0.8 m in front of the last obstacle and up to 1.4 m in front of the exit doors. The agents also get closer to each other when they are members of movement groups, as in Scenario 3. Hence, the number of new infected agents also increases for that scenario. The detailed connections are explained in the connected individual results chapter 7.2.1.3.

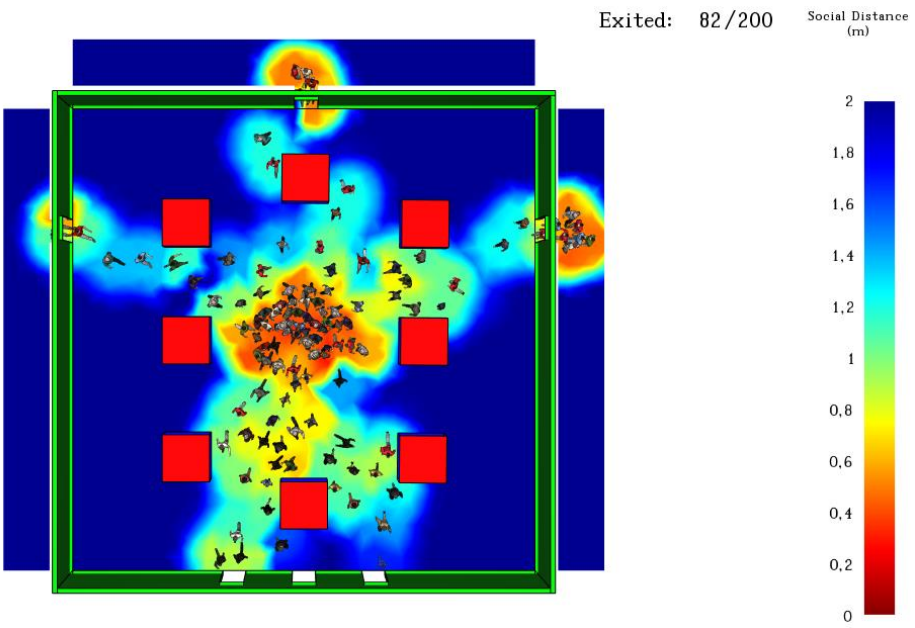

Figure 31 - Scenario 1 - Social distance of agents, Screenshot Pathfinder

Furthermore, the movement paths of the agents in Figure 31 shows the influence of the obstacles in the model. From the left- and right-side entrances, the agents only move on one side of the obstacles. The agent flow originating from the top entrance splits into two halves, walking on both sides of the closest obstacle – the same is the case in front of the exits. That explains the differences between the public space model without and with obstacles. For that purpose, the accumulated area usages of both model versions are presented in Figure 32.

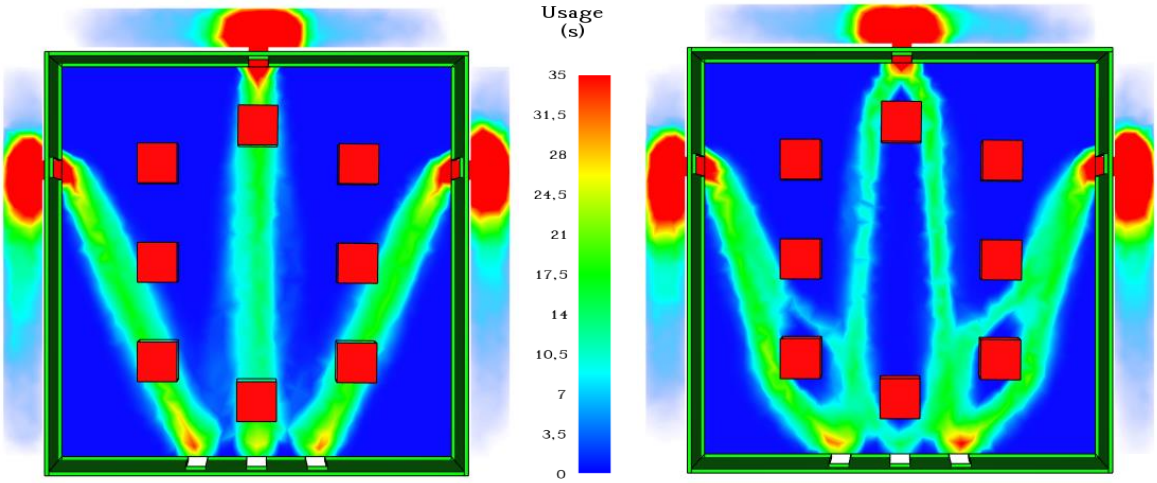

Figure 32 - Accumulated usage, l. without, r. with obstacles, Screenshots Pathfinder

In the model on the left, the agents can walk through the obstacles, only shown for orientation. Therefore, all agents take the direct way from the entrance to the closest exit. The resulting accumulated usage is almost equal along their way and increases in front of the doors. When the obstacles are physical hurdles, the agents must move around them. The agent flow in these areas and the resulting accumulated usage are reduced (Figure 32 right). The obstacle in front of the middle exit door causes the agents to preferably use the other two exits, decreasing the usage in the middle and increasing it on the sides. When avoiding the obstacles, the agents get closer to each other, increasing the likelihood of

(close) contacts and, therefore, the active infection probability. That finally increases the number of new infected agents. The tendency is counteracted by the split agent flow but results in, on average, slightly higher numbers of new infected agents.

On the other side, the number of new infected agents is lower in Scenario 2 when changing the demographics to an older population. According to the agent attributes stated in Table 2 and Table 5, older agents move slower and exhale their breathing air less forcefully. Both result in smaller infection areas for I-agents, making contact with S-agents less likely. This correlation outweighs the one of the obstacles and causes the lower infection numbers.

In the minimal model, the high relative number of new infected agents is justified by comparing the number of agents and the size of the evaluated room. The minimal model contains 20 agents in a room of 25 m² (square with sides of 5 m each), resulting in a theoretical population density of 20 / 25 = 0.80 agents per square meter. Analogously, the 200 agents and room size of 400 m² indicate a population density of 0.50 agents per square meter for the public space model. In addition with the only two, instead of three, entrance doors in the minimal model, the higher population density means less space to avoid contact with other agents, increasing the contact number and infection probability.

In all models and scenarios, the active infection mechanism dominates the passive one, as the passive infection probability remains at the lower limit of 5 %.

8 Conclusion

8.1 Conclusion and discussion

Referring back to the scope of this report (chapter 2.3), this chapter answers the research question, controlling the fulfilment of the stated objectives. Table 10 shows an overview containing short descriptions of all main and sub-objectives, evaluating their fulfilment. The two main objectives are underlined. The degree of fulfilment is expressed by three options: entire, partial and no. Entire fulfilment meets the objective without known restraints – No fulfilment results from complete objective failure. Fundamentally achieved objectives with only minor shortcomings are partially fulfilled. Additionally, this option is applied when the entire fulfilment is questionable.

Objective		Fulfilment degree		
Nr.	Short description	Entire	Partial	No
1	<u>Develop a model to estimate transmission rate</u>	X		
1.1	Consider the current state of the art	X		
1.2	Include active and passive transmission		X	
1.3	Ensure flexibility and adaptability of the model	X		
1.4	Ensure reproducibility of the model		X	
2	<u>Test the model analysing local conditions</u>	X		
2.1	Select relevant local conditions	X		
2.2	Point out potential future research topics		X	

Table 10 - Model evaluation - Fulfilment of objectives

As shown in the table, the first objective to develop a model to estimate transmission rates of airborne diseases is fulfilled entirely. The model is created and functional, delivering the number of new infected agents as a description of the transmission rate. Furthermore, it is based on the current state of the art (sub-objective 1.1) and is flexible enough to be adapted to the different scenarios (sub-objective 1.3). The sub-objective number 1.2, including active and passive transmission mechanisms, is only partly fulfilled since it accounts for both mechanisms, but the resulting passive infection probability does not change between the scenarios. Given that a low passive infection probability is plausible for short exposure times, that must not be an error in the model but does not prove the functionality of the passive mechanism neither. The ODD protocol emphasises the importance of ensuring information accessibility for the reader. It is critical because the modelling approach is a prepared framework for other modellers (sub-objective 1.4). The sub-objective is only partly fulfilled because it is on the reader to decide. Furthermore, it requires significant effort to rebuild the model.

The second main objective, to test the model analysing relevant local conditions, is entirely fulfilled as the changes of the room (size and adding obstacles) and in the scenarios are modelled and evaluated successfully, additionally fulfilling sub-objective 2.1. The observed correlations are that larger rooms and older populations reduce disease transmission. On the other side, adding obstacles to the room, agents moving in groups or gathering and waiting in a part of the room increase disease transmission. All these observations are plausible, as explained in chapter 7.3. It is to note that the model does not include the influence the state of the agents' immune system has on their infection risks. Otherwise, the infection risk of an older population may be higher than the one of younger people. That is related to

the last sub-objective (number 2.2) and the next chapter of this report (chapter 8.2). Considering the complexity of airborne disease transmission modelling, the list of suggested topics for future research can never be exclusive, but the one in chapter 8.2 can be a starting point.

Concluding the fulfilment of the objectives, the purpose of the model and report are achieved. The model may not be perfect so far, but it offers a functional framework for further development and adaptation to real places and diseases, e. g., COVID-19.

8.2 Future work

Airborne diseases are complex and, especially considering the current COVID-19 pandemic, a valuable topic to research in – the same counts for disease transmission modelling in specific. It would be helpful for the responsible decision-makers if they could use reliable computer simulations to pre-estimate the outcomes of potential disease transmission countermeasures. Therefore, a next step could be to modify the proposed model to fit the specifics of COVID-19 or other current diseases. Once a related model is created, it could quickly be adapted to fit new variants, like the Delta and Omicron variants of COVID-19. Besides, if communicated to the general public, the use of pre-evaluated measures may result in a higher acceptance of those.

The modelled understanding of disease transmission could be expanded, considering other scenarios or additional attributes – one example from the conclusion: the influence of the agents' immune system depending on his age. A possible way of implementing this attribute is to add it to the agent profile, similar to the age-dependent distance modification factor. A placeholder is implemented in the connected programming Script 0 to highlight this option (appendix 12.3.1). Other considerable attributes are, e. g., the usage of different kinds of face masks and the reduced number of infected agents caused by entrance restrictions. Experienced programmers may find options to optimise the programming parts without changing the model itself.

With these improvements and more powerful computers, the model could be used to analyse large-scale environments over long periods and could be very valuable to look at for decision-makers.

9 List of figures

10 List of tables

11 References

Al-Azawi, A., Nørregard Petersen, J., Hennings, M., Ledwaba, S.C., Chornobryvets, S., 2021. Evacuation of passengers from a Train in the Fehmarnbelt Fixed Link (Student report). Aalborg University Esbjerg.

Aliabadi, A.A., Rogak, S.N., Bartlett, K.H., Green, S.I., 2011. Preventing Airborne Disease Transmission: Review of Methods for Ventilation Design in Health Care Facilities. Advances in Preventive Medicine 2011, 1–21. https://doi.org/10.4061/2011/124064

Alshammari, S.M., Ba-Aoum, M.H., Alganmi, N.A., Allinjawi, A.A., 2021. Agent-Based Simulation Framework for Epidemic Forecasting during Hajj Seasons in Saudi Arabia. Information 12, 325. https://doi.org/10.3390/info12080325

Auchincloss, A.H., Garcia, L.M.T., 2015. Brief introductory guide to agent-based modeling and an illustration from urban health research. Cad. Saúde Pública 31, 65–78. https://doi.org/10.1590/0102-311X00051615

Burrows, W., Scarpelli, Dante, G., 2020. Disease [WWW Document]. Encyclopedia Britannica. URL https://www.britannica.com/science/disease (accessed 1.6.22).

CDC, C. for D.C. and P., 2012. Principles of Epidemiology in Public Health Practice (Self-study No. Third Edition). Atlanta, USA.

Cuevas, E., 2020. An agent-based model to evaluate the COVID-19 transmission risks in facilities. Computers in Biology and Medicine 121, 103827. https://doi.org/10.1016/j.compbiomed.2020.103827

DeWitte, S.N., 2014. Mortality Risk and Survival in the Aftermath of the Medieval Black Death. PLoS ONE 9, e96513. https://doi.org/10.1371/journal.pone.0096513

D'Orazio, M., Bernardini, G., Quagliarini, E., 2021a. A probabilistic model to evaluate the effectiveness of main solutions to COVID-19 spreading in university buildings according to proximity and time-based consolidated criteria. Build. Simul. 14, 1795–1809. https://doi.org/10.1007/s12273-021-0770-2

D'Orazio, M., Bernardini, G., Quagliarini, E., 2021b. Sustainable and resilient strategies for touristic cities against COVID-19: An agent-based approach. Safety Science 142, 105399. https://doi.org/10.1016/j.ssci.2021.105399

EC, E.C., 2019. Beginners:GDP - What is gross domestic product (GDP)? [WWW Document]. Eurostat - statistics explained. URL https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Beginners:GDP_-_What_is_gross_domestic_product_(GDP)?#What.E2.80.99s_covered_by_GDP.3F (accessed 12.30.21).

Flores, M.V., Cohen, M., 2014. Preventing Airborne Disease Transmission: Implications for Patients During Mechanical Ventilation, in: Esquinas, A.M. (Ed.), Noninvasive Ventilation in High-Risk Infections and Mass Casualty Events. Springer Vienna, Vienna, pp. 305–313. https://doi.org/10.1007/978-3-7091-1496-4_34

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S.K., Huse, G., Huth, A., Jepsen, J.U., Jørgensen, C., Mooij, W.M., Müller, B., Pe'er, G., Piou, C., Railsback, S.F., Robbins, A.M., Robbins, M.M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R.A., Vabø, R., Visser, U., DeAngelis, D.L., 2006. A standard protocol for describing individual-based and agent-based models. Ecological Modelling 198, 115–126. https://doi.org/10.1016/j.ecolmodel.2006.04.023

Huremović, D. (Ed.), 2019. Psychiatry of pandemics: a mental health response to infection outbreak. Springer, Cham.

Jarus, O., 2021. 20 of the worst epidemics and pandemics in history [WWW Document]. Live Science. URL https://www.livescience.com/worst-epidemics-and-pandemics-in-history.html (accessed 12.30.21).

Katella, K., 2021. Our Pandemic Year—A COVID-19 Timeline [WWW Document]. Yale Medicine. URL https://www.yalemedicine.org/news/covid-timeline (accessed 12.30.21).

Laskowski, M., Demianyk, B.C.P., Witt, J., Mukhi, S.N., Friesen, M.R., McLeod, R.D., 2011. Agent-Based Modeling of the Spread of Influenza-Like Illness in an Emergency

Department: A Simulation Study. IEEE Trans. Inform. Technol. Biomed. 15, 877–889. https://doi.org/10.1109/TITB.2011.2163414

Mao, L., 2015. Predicting Self-Initiated Preventive Behavior Against Epidemics with an Agent-Based Relative Agreement Model. JASSS 18, 6. https://doi.org/10.18564/jasss.2892

Perez, L., Dragicevic, S., 2009. An agent-based approach for modeling dynamics of contagious disease spread. Int J Health Geogr 8, 50. https://doi.org/10.1186/1476-072X-8-50

Pierce, R., 2021. Kite Definition [WWW Document]. Math is fun. URL https://www.mathsisfun.com/definitions/kite.html (accessed 12.5.21).

Pierce, R., 2018. Definition of Octagon [WWW Document]. Math is fun. URL http://www.mathsisfun.com/definitions/octagon.html (accessed 12.5.21).

Serafini, G., Parmigiani, B., Amerio, A., Aguglia, A., Sher, L., Amore, M., 2020. The psychological impact of COVID-19 on the mental health in the general population. QJM: An International Journal of Medicine 113, 531–537. https://doi.org/10.1093/qjmed/hcaa201

Szmigiera, M., 2021. Impact of the coronavirus pandemic on the global economy - Statistics & Facts [WWW Document]. Statista. URL https://www.statista.com/topics/6139/covid-19-impact-on-the-global-economy/#dossierKeyfigures (accessed 12.30.21).

Thunderhead, T.E.C.Inc., 2019. Pathfinder [WWW Document]. Thunderhead Engineering. URL https://www.thunderheadeng.com/pathfinder/ (accessed 12.11.21).

Tracy, M., Cerdá, M., Keyes, K.M., 2018. Agent-Based Modeling in Public Health: Current Applications and Future Directions. Annu. Rev. Public Health 39, 77–94. https://doi.org/10.1146/annurev-publhealth-040617-014317

WHO, W.H.O., 2021. Coronavirus disease (COVID-19) [WWW Document]. WHO. URL https://www.who.int/health-topics/coronavirus (accessed 11.13.21).

WHO, W.H.O., 2019. Pandemic influenza [WWW Document]. WHO Regional Office for Europe. URL https://www.euro.who.int/en/health-topics/communicable-diseases/influenza/pandemic-influenza (accessed 12.3.21).

WHO, W.H.O., n.d. Past pandemics [WWW Document]. WHO Regional Office for Europe. URL https://www.euro.who.int/en/health-topics/communicable-diseases/influenza/pandemic-influenza/past-pandemics (accessed 12.30.21).

Zhou, Y., Nikolaev, A., Bian, L., Lin, L., Li, L., 2021. Investigating transmission dynamics of influenza in a public indoor venue: An agent-based modeling approach. Computers & Industrial Engineering 157, 107327. https://doi.org/10.1016/j.cie.2021.107327

12 Appendix

12.1 Detailed literature review procedure

KEYWORDS GROUPS

1. Disease related keyword
    - Disease / diseases
    - Disease transmission
    - Airborne disease
    - Airborne disease transmission
2. Agent-based modelling related keywords
    - Agent-based modeling
    - Agent-based modelling
    - ABM
    - Occupancy modelling
    - Occupancy modeling
    - Non-evacuation behavior
    - Non-evacuation behaviour
    - Evacuation behavior
    - Evacuation behaviour
    - Evacuation simulation
    - Egress simulation
3. Methods/tools related keywords
    - Probabilistic
    - Safety
    - Risk
    - Probability
    - Simulation

SEARCH QUERIES

Number	ID	Query	Results
1	Q1	TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")	4,766,941
2	Q2	TS= ("Agent-based modeling" OR "Agent-based modelling" OR "ABM" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation")	10,785
3	Q3	TS=("probabilistic" OR "safety" OR "risk" OR "probability" OR "simulation")	6,686,796
4	Q1+Q2	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS= ("Agent-based modeling" OR "Agent-based modelling" OR "ABM" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation")	624
5	Q1+Q3	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS=("probabilistic" OR "safety" OR "risk" OR "probability" OR "simulation")	1,192,182
6	Q1+Q2+Q3	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS= ("Agent-based modeling" OR "Agent-based modelling" OR "ABM" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation") AND TS=("probabilistic" OR "safety" OR "risk" OR "probability" OR "simulation")	320

CLUSTERING ANALYSIS

NUMBER: 4 M

FINAL QUERY: Q1 + Q2

RESULTS previous: 624

4 correction	Q1+Q2	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS= ("Agent-based modeling" OR "Agent-based modelling" OR "ABM" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation") NOT TS=("mosquito" OR "mosquitos")	
4 modified	Q1+Q2	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS= ("Agent-based modeling" OR "Agent-based modelling" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation") NOT TS=("mosquito" OR "mosquitos")	336

CORRECTION EXPLANATION:

The abbreviation "ABM" for agent-based modelling is excluded because the cluster analysis identified that it is also used in many different contexts out of which agent-based modelling is not dominating the others. This term is, therefore, taken out of the search query.

The first cluster analysis furthermore shows that several references address diseases spread by mosquitos. Since these are not considered relevant in this report, they are excluded from the relevant literature.

NUMBER: 6 M1

FINAL QUERY: Q1 + Q2 + Q3

RESULTS previous: 320

6 correction 1	Q1+Q2+Q3	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS= ("Agent-based modeling" OR "Agent-based modelling" OR "ABM" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation") AND TS=("probabilistic" OR "safety" OR "risk" OR "probability" OR "simulation") NOT TS=("mosquito" OR "mosquitos")	
6 modified 1	Q1+Q2+Q3	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS= ("Agent-based modeling" OR "Agent-based modelling" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation") AND TS=("probabilistic" OR "safety" OR "risk" OR "probability" OR "simulation") NOT TS=("mosquito" OR "mosquitos")	224

CORRECTION EXPLANATION:

Here, the changes descripted above in query number 4 M are implemented for the combination of all three keyword groups.

NUMBER: 6 M2

FINAL QUERY: Q1 + Q2 + Q3

RESULTS previous: 224

6 modified 2	Q1+Q2+Q3	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS= ("Agent-based modeling" OR "Agent-based modelling" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation") AND TS=("probabilistic" OR "safety" OR "risk" OR "probability" OR "simulation") NOT TS=("mosquito" OR "mosquitos") NOT TS=("abm" OR "abms")	
6 modified 2	Q1+Q2+Q3	(TI=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission") OR TS=("disease" OR "diseases" OR "disease transmission" OR "airborne disease" OR "Airborne disease transmission")) AND TS= ("Agent-based modeling" OR "Agent-based modelling" OR "Occupancy modelling" OR "Occupancy modeling" OR "Non-evacuation behavior" OR "Non-evacuation behaviour" OR "Evacuation behavior" OR "Evacuation behaviour" OR "Evacuation simulation" OR "Egress simulation") AND TS=("probabilistic" OR "safety" OR "risk" OR "probability" OR "simulation") NOT TS=("mosquito" OR "mosquitos") NOT TS=("abm" OR "abms")	193

CORRECTION EXPLANATION:
The cluster analysis shows that after excluding the search keyword "abm", there are still many references with a focus on other abm definitions. To avoid this, the abbreviations "abm" and "abms" are explicitly excluded from the search to set the correct focus on agent-based modelling.

12.2 Individual models in state of the art literature

Literature	Objective	Basic model description	
		Agent characterisation	Simulation / Calculation of infections
(D'Orazio et al., 2021a)	Compare effectiveness of COVID-19 measurements to identify the optimal economic-social balance in universities	Individual agents are characterised by their epidemical status (SI), wearing or not what kind of face mask, and being or not asymptomatic in case of an infection.	The modelled world is divided into lecture and break areas. Each agent has a random position in every area he is using following the daily schedule. Within an area, this position is fixed (agents return to the same position when returning to a room). At each time step, S-agents within a distance of ≤ 2 m to an I-agent become infected with a probability Pvir dependent on the exposure time, the mask-wearing and mask protection of both agents, the time passed after the infection of the I-agent, and the overall immunity of the simulated population.
(Laskowski et al., 2011)	Develop a framework to simulate the virus transmission in a hospital emergency department	Individual agents are first divided into the two groups healthcare workers and patients and further characterised by their epidemical status (SI+ immune) and treatment priority.	For the agent movement simulation, a layout plan of a specific emergency department and observed typical procedures (registration - waiting room - treatment) are used. For each time step, S-agents become infected with general probabilities if they are in close or casual contact to I-agents.
(Alshammari et al., 2021)	Propose a framework to predict the epidemic development considering mass gatherings during the Hajj season	Individual agents are defined and grouped by several specifics of the categories Demographic, Travel-related, Epidemic-related (SEIR), and Hajj-related influencing the individual transmission and infection probabilities.	The calculation is based on a set of behavioural rules resulting in contact time within a group and between different groups leading to individual infection probabilities at the different phases of the pilgrimage from arrival to departure. The final epidemical status is simulated based on these probabilities.
(Perez and Dragicevic, 2009)	Simulate disease transmission in an urban environment considering typical agent mobility within the city	Individual agents characterised by their epidemical status (SEIR) and typical activities and movement within the city.	The agents follow their normal daily activities. If an S-agent gets in close contact (≤ 1 m distance) with an I-agent, he becomes an E-agent with a general probability and becomes infectious after a certain time period. After a certain time, I-agents recover and cannot become infected again. The agents' movement and transmission is tracked to be presented on a map to identify and highlight transmission hotspots.

(Cuevas, 2020)	Evaluate COVID-19 transmission risks in facilities	Individual agents are divided into the two groups I and S and assigned uniformly distributed infection (Pri) and mobility (Prcm) probabilities.	The agents are randomly placed in a coordinate system. S-agents become infected with Pri if they are within the circular area (radius R) of an infected agent. Afterwards, each agent moves a random distance S with Prcm and the simulation continues with the next iteration.
(Zhou et al., 2021)	Create a framework for disease transmission in public indoor venues	Individual agents are grouped into customers and employees and characterised by several attributes within the categories demographics (individual and groups), mobility (within a location and between locations), and epidemical status (SEIR).	The movement respectively the number and duration of contacts of both agent groups is simulated based on observed behaviour. If an S-agent gets into contact with an I-agents (distance ≤ 1.6 m), he becomes an E-agent general probability per exposure time. While the customers only visit the venue for a short time period, the employees are there every day and, therefore, become I-agents after a certain time period.
(Mao, 2015)	Analyse the connections between epidemic transmission and individual self-initiated behaviour	Individual agents are grouped into, e. g., families and workplaces. Agents are characterised by epidemical status (SLIR) and preventive behaviour attitude (positive, neutral, or negative).	The simulation starts with one infected agent to spread the disease and one informed agent to spread information about preventive behaviour. All others are S-agents. Disease transmission occurs at each simulation step based on a transmission rate that can be reduced by preventive behaviour of the agents. The likelihood of adopting preventive behaviour depends on the agents' attitude towards it. The attitude changes with a probability if an S-agent becomes infected. If an agent becomes infected, he receives a latent status in which he cannot spread the disease. I-agents can only transmit the disease for a certain time before they become R-agents.
(D'Orazio et al., 2021b)	Compare effectiveness of COVID-19 measurements to identify the optimal economic-social balance in touristic areas	Individual agents are characterised by origin (resident or tourist) and epidemical status (SI).	The agents are placed with random uniformly distributed distances (d) between 1 m and 3 m. S-agents can become infected with a general infection probability based on the mask-wearing and mask filter percentage if d to an I-agent ≤ 2 m. Afterwards, each agent moves a random distance, and the simulation continues with the next iteration. New I-agents have an initially lower transmission probability increasing with the time after their own infection. After a delay period, I-agents can be asymptomatic or non-asymptomatic. Non-asymptomatic agents can be removed from the model.

Table 11 - Individual models in state of the art literature

12.3 Programming code

12.3.1 Script 0 – Agent profile parameters

```
## remove variables in the main environment
rm(list=ls())

# Import input from Pathfinder output
name_of_profiles_file   = "3D 5x5 RUN_profiles_occupant_params.csv"
path_input_file     = paste(getwd(),"/",sep="")
input_file_profiles     = paste(path_input_file, name_of_profiles_file, sep="" )
profiles_info = read.csv(file = input_file_profiles, skip=0, header = TRUE)
max_num_agents = max(profiles_info$id) + 1

# Create output table
output_table = data.frame(profiles_info$id, profiles_info$profile, pro-
files_info$maxVel.m.s.)
colnames (output_table) = c("id", "profile", "v_max")

output_table

# Create list of profiles
p_infec_dist = array(1, length(max_num_agents))
p_infec_prob = array(1, length(max_num_agents))

list_profiles = data.frame(sort(unique(output_table$profile)), p_infec_dist,
p_infec_prob)
colnames (list_profiles) = c("profile", "p_infec_area", "p_infec_prob")

list_profiles

n_profiles = length(list_profiles$profile)
n_profiles

# Assign profile specific parameters

# Specify active infection distance
for (m in 1:n_profiles) {
if      (grepl("2-5"       , list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 0.25}
else if (grepl("6-9"  , list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 0.50}
else if (grepl("10-19", list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 1.00}
else if (grepl("20-29", list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 1.00}
else if (grepl("30-39", list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 1.00}
else if (grepl("40-49", list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 1.00}
else if (grepl("50-59", list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 0.75}
else if (grepl("60-69", list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 0.75}
else if (grepl("70+"  , list_profiles$profile[m], fixed = TRUE) == TRUE)
{list_profiles$p_infec_area[m] = 0.50}
else {list_profiles$p_infec_area[m] = "F"}
}

# Specify infection probability - PLACEHOLDER

list_profiles
# Add list of profiles to output table
output_p_infec_dist = array(NaN, max_num_agents)
output_p_infec_prob = array(NaN, max_num_agents)

for (n in 1:max_num_agents) {
for (m in 1:n_profiles) {
if (output_table$profile [n] == list_profiles$profile [m]) {
output_p_infec_dist [n] = list_profiles$p_infec_dist [m]
```

```
output_p_infec_prob [n] = list_profiles$p_infec_prob [m]
}
}
}

# Select infected agents
# status = S = susceptible agent
status = array("S", dim = max_num_agents)

# list all id of infected agents
## 0 = give a vector
## 1 = simulated the infected according to %
option_percentage  = 1 # 1 = Yes
perc_infect_agent  = 50
perc_recov_agent   = 5
perc_vacc_agent    = 10

# All other agents are susceptible

if (option_percentage == 0) {
infected_agents   = c(0, 2, 4, 6, 8, 10, 12, 14, 16)
recovered_agents  = c(1)
vaccinated_agents = c(2, 5)
} else {
vec_agent  = sample(seq(0, max_num_agents, by=1) )
infected_agents   = vec_agent[1 : ceiling((perc_in-
fect_agent/100)*max_num_agents)]
recovered_agents  = vec_agent[(length(infected_agents) + 1) : (length(in-
fected_agents) + ceiling((perc_recov_agent/100)*max_num_agents))]

vec_agent  = sample(seq(0, max_num_agents, by=1) )
vaccinated_agents = vec_agent[1 : ceiling((perc_vacc_agent/100)*max_num_agents)]
}

# Assign agent status
# infected
for (k in 1:max_num_agents) {
if (is.element(output_table$id[k], infected_agents)) {
if (is.element(output_table$id[k], vaccinated_agents)) {
status [k] = "VI"
}
else {status [k] = "I"}
}
}

# recovered
for (k in 1:max_num_agents) {
if (is.element(output_table$id[k], recovered_agents)) {
status [k] = "R"
}
}

# vaccinated
for (k in 1:max_num_agents) {
if (is.element(output_table$id[k], vaccinated_agents)) {
if (status[k] == "VI") {}
else {status [k] = "V"}
}
}

# Select mask-wearing agents
option_percentage = 1
perc_masked_agent = 10

masked = array(0, max_num_agents)

if (option_percentage == 0) {masked_agents = c(0, 4, 6)
} else {
vec_agent  = sample(seq(0, max_num_agents, by = 1) )
masked_agents = vec_agent[1 : ceiling((perc_masked_agent/100)*max_num_agents)]
```

```
}

for (k in 1:max_num_agents) {
if (is.element(output_table$id[k], masked_agents)) {
masked [k] = "X"
}
}

# Complete output table
output_table = cbind(output_table, status, masked, output_p_infec_dist, out-
put_p_infec_prob)
colnames (output_table) = c("id", "profile", "v_max", "status", "masked", "p_in-
fec_dist", "p_infec_prob")

# Show results
output_table

# Export results
export = 0 # export if 1, do nothing if 0

if (export == 1) {
name_output_file = "Agent parameters.txt"
path_output_file = path_input_file
output_file      = paste(path_output_file, name_output_file, sep="" )
write.table (output_table, file = output_file, row.names = FALSE, col.names =
TRUE, sep = "\t")

}
```

12.3.2 Script 1 – Interpolation and infection areas

```
## remove variables in the main environment
rm(list=ls())

# Import input from Pathfinder output
name_of_details_file = "3D 5x5 RUN_profiles_occupants_detailed.csv"
path_input_file = paste(getwd(),"/",sep="")
input_file_details = paste(path_input_file, name_of_details_file , sep="" )
agents_info   = read.csv(file = input_file_details , skip=2, header = TRUE)
colnames(agents_info) = c("time","id","name","ac-
tive","x","y","z","vel","dist","loc","terrain","lgs")
max_num_agents = max(agents_info$id) + 1

# Exclude agents in waiting areas
# Empty location (room) may occur for the last position ("in" the exit door)
agents_info = agents_info[agents_info$loc == "Floor" | agents_info$loc == "",]

# Import agent profile parameters from A0
name_of_profiles_file   = "Agent parameters.txt"
input_file_profiles     = paste(path_input_file, name_of_profiles_file, sep="" )
profiles_info = read.table(file = input_file_profiles, skip=0, header = TRUE)

# Define function for linear movement parameters
linear_eq = function(val,opt,x1,y1,x2,y2){
m = (y2 - y1) / (x2 - x1)
if (opt == "y"){  # delivering y(x)
linear_eq <- m*(val-x1) + y1
} else if (opt == "x") { # delivering x(y)
linear_eq = ((val - y1)/m) + x1
} else if (opt == "m") { # slope
linear_eq = m
}
}

# Interpolation for all agents
# Smallest time interval
delta_t_min = 0.25

# Calculated m based on standard Pathfinder time interval = 1 s
m = 1/delta_t_min
temp = array (0, dim = (m+1))

# Calculate infectious area around infected agent (rectangular shape)
d_front_breathing    = 0.8  # m
alpha_front_breathing = 0.5  # percentage

alpha_back = 3.5 # times d_front_breathing

# infection area factor for infected agents wearing face masks
area_red_masks = 0.2

# Max velocity based on demographics, site conditions, etc
max_contextual_vel = max(profiles_info$v_max)  # m/s

# Create empty output table
output_table = data.frame(matrix(ncol = 22,nrow = 0))

for (j in 0: max(agents_info$id)) {

# Selection of agent

time_agent = agents_info$time[ agents_info$id == j ]
x_agent    = agents_info$x[ agents_info$id == j ]
y_agent    = agents_info$y[ agents_info$id == j ]
v_agent    = agents_info$vel[ agents_info$id == j ]

status_agent = profiles_info$status [j+1]
masked_agent = profiles_info$masked [j+1]
```

```
# Assign agent parameters from A0
p_infec_dist = profiles_info$p_infec_dist [profiles_info$id == j]

# Assign side distance of infection area
d_s = p_infec_dist * 0.50# sides

# Creation of new arrays

id_agent_new   = array(j, dim=c( ((length(x_agent)-1)*m+1) ))
time_agent_new = array(NaN, dim=c( ((length(x_agent)-1)*m+1) ))
x_agent_new    = array(0, dim=c( ((length(x_agent)-1)*m+1) ))
y_agent_new    = array(0, dim=c( ((length(y_agent)-1)*m+1) ))
v_agent_new    = array(0, dim=c( ((length(v_agent)-1)*m+1) ))

time_agent_new [length(time_agent_new)] = 0

# For each infected agent
write_row = 1

for (k in 1:(length(time_agent)-1)){
m_temp = m
if (time_agent[k+1] - time_agent [k] < 1) {
m_temp = floor((time_agent[k+1] - time_agent [k])/delta_t_min)
}

temp [1:(m_temp+1)] = seq(time_agent[k], time_agent[k+1], length.out =
(m_temp+1))
time_agent_new [write_row:(write_row + m_temp - 1)] = temp [1:m_temp]

temp [1:(m_temp+1)] = seq(x_agent[k], x_agent[k+1], length.out = (m_temp+1))
x_agent_new    [write_row:(write_row + m_temp - 1)] = temp [1:m_temp]

temp [1:(m_temp+1)] = seq(y_agent[k], y_agent[k+1], length.out = (m_temp+1))
y_agent_new    [write_row:(write_row + m_temp - 1)] = temp [1:m_temp]

temp [1:(m_temp+1)] = seq(v_agent[k], v_agent[k+1], length.out = (m_temp+1))
v_agent_new    [write_row:(write_row + m_temp - 1)] = temp [1:m_temp]

write_row = write_row + m_temp
}

# Exclude "empty" rows
# Manual loop since "for" loop doesn't allow stepping backwards
loop_step = 1
loop_end  = length(time_agent_new)

# Don't delete the last row to avoid overwriting by final values
time_agent_new [length(time_agent_new)] = 0

while (loop_step < loop_end) {
if (is.nan(time_agent_new[loop_step])) {
time_agent_new = time_agent_new[-loop_step]
x_agent_new = x_agent_new[-loop_step]
y_agent_new = y_agent_new[-loop_step]
v_agent_new = v_agent_new[-loop_step]
id_agent_new = id_agent_new [-loop_step]
loop_step = loop_step - 1
}
loop_step = loop_step + 1
}

# For each agent - Add final values
time_agent_new [length(time_agent_new)] = time_agent[length(time_agent)]
x_agent_new [length(x_agent_new)] = x_agent[length(x_agent)]
y_agent_new [length(y_agent_new)] = y_agent[length(y_agent)]
v_agent_new [length(v_agent_new)] = v_agent[length(v_agent)]

# Define distances infected area for casual contact
d_x_f_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_x_b_active_inf_area = array(0, dim=c(length(time_agent_new)))
```

```
d_x_r_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_x_l_active_inf_area = array(0, dim=c(length(time_agent_new)))

d_y_f_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_y_b_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_y_r_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_y_l_active_inf_area = array(0, dim=c(length(time_agent_new)))

# Expended for additional middle points to create octagon infection area
d_M_f_r_x_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_M_f_l_x_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_M_b_r_x_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_M_b_l_x_active_inf_area = array(0, dim=c(length(time_agent_new)))

d_M_f_r_y_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_M_f_l_y_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_M_b_r_y_active_inf_area = array(0, dim=c(length(time_agent_new)))
d_M_b_l_y_active_inf_area = array(0, dim=c(length(time_agent_new)))

if (status_agent == "I" | status_agent == "VI") {

# Calculate slope for each line between to points
slope_vector_m = array(0, dim=c(length(time_agent_new)-1))

slope_vector_rad = array(0, dim=c(length(time_agent_new)-1))
slope_vector_deg = array(0, dim=c(length(time_agent_new)-1))

for(z in 1:(length(time_agent_new)-1)){
slope_vector_m[z] = linear_eq(0,'m',x_agent_new[z],y_agent_new[z],
x_agent_new[z+1],y_agent_new[z+1])

slope_vector_rad[z]= atan(slope_vector_m [z])

slope_vector_deg[z] = slope_vector_rad[z]*180/pi
}

# Calculate edge points
for (l in 1:(length(time_agent_new)-1)) {
## F and B dependent on the movement direction
gamma_parameter = rnorm(1, mean=1, sd=0.05)
d_f = p_infec_dist * linear_eq(v_agent_new[l],'x',(d_front_breathing*al-
pha_front_breathing),
max_contextual_vel,
d_front_breathing,0)*gamma_parameter
d_b = linear_eq(v_agent_new[l],'x', 0,0,
 d_f*alpha_back,max_contextual_vel)

# Influence of face masks
if (masked_agent == "X") {
d_f = area_red_masks * d_f
d_b = area_red_masks * d_b
}

# Determination of mathematical sign
d_x_r_active_inf_area [l] = + sin(slope_vector_rad[l])*d_s
d_x_l_active_inf_area [l] = - sin(slope_vector_rad[l])*d_s

d_y_r_active_inf_area [l] = - cos(slope_vector_rad[l])*d_s
d_y_l_active_inf_area [l] = + cos(slope_vector_rad[l])*d_s

if (x_agent_new [l+1] > x_agent_new [l]) {
d_x_f_active_inf_area [l] = + cos(slope_vector_rad[l])*d_f
d_x_b_active_inf_area [l] = - cos(slope_vector_rad[l])*d_b
d_y_f_active_inf_area [l] = + sin(slope_vector_rad[l])*d_f
d_y_b_active_inf_area [l] = - sin(slope_vector_rad[l])*d_b
}
else if (x_agent_new [l+1] < x_agent_new [l]) {
d_x_f_active_inf_area [l] = - cos(slope_vector_rad[l])*d_f
d_x_b_active_inf_area [l] = + cos(slope_vector_rad[l])*d_b
d_y_f_active_inf_area [l] = - sin(slope_vector_rad[l])*d_f
```

```
d_y_b_active_inf_area [l] = + sin(slope_vector_rad[l])*d_b
}
else  {
if (y_agent_new [l+1] != y_agent_new [l]) {
d_x_f_active_inf_area [l] = + cos(slope_vector_rad[l])*d_f
d_x_b_active_inf_area [l] = - cos(slope_vector_rad[l])*d_b
d_y_f_active_inf_area [l] = + sin(slope_vector_rad[l])*d_f
d_y_b_active_inf_area [l] = - sin(slope_vector_rad[l])*d_b
}

else {# Neither x- nor y-coordinates changed -> position of agent unchanged ->
active infection area unchanged
d_x_f_active_inf_area [l] = d_x_f_active_inf_area [l-1]
d_x_b_active_inf_area [l] = d_x_b_active_inf_area [l-1]
d_y_f_active_inf_area [l] = d_y_f_active_inf_area [l-1]
d_y_f_active_inf_area [l] = d_y_f_active_inf_area [l-1]

# Re-assign coordinates of side points which are empty (NA) for no movement
d_x_r_active_inf_area [l] = d_x_r_active_inf_area [l-1]
d_x_l_active_inf_area [l] = d_x_l_active_inf_area [l-1]

d_y_r_active_inf_area [l] = d_y_r_active_inf_area [l-1]
d_y_l_active_inf_area [l] = d_y_l_active_inf_area [l-1]
}
}

# Addtional points for octagond_M_f_l_x_active_inf_area [l] = 1.2 * d_x_l_ac-
tive_inf_area [l] + (1.2 * d_x_f_active_inf_area [l] - 1.2 * d_x_l_ac-
tive_inf_area [l]) / 2
d_M_f_r_x_active_inf_area [l] = 1.2 * d_x_r_active_inf_area [l] + (1.2 *
d_x_f_active_inf_area [l] - 1.2 * d_x_r_active_inf_area [l]) / 2
d_M_b_l_x_active_inf_area [l] = 1.2 * d_x_l_active_inf_area [l] + (1.2 *
d_x_b_active_inf_area [l] - 1.2 * d_x_l_active_inf_area [l]) / 2
d_M_b_r_x_active_inf_area [l] = 1.2 * d_x_r_active_inf_area [l] + (1.2 *
d_x_b_active_inf_area [l] - 1.2 * d_x_r_active_inf_area [l]) / 2

d_M_f_l_y_active_inf_area [l] = 1.2 * d_y_l_active_inf_area [l] + (1.2 *
d_y_f_active_inf_area [l] - 1.2 * d_y_l_active_inf_area [l]) / 2
d_M_f_r_y_active_inf_area [l] = 1.2 * d_y_r_active_inf_area [l] + (1.2 *
d_y_f_active_inf_area [l] - 1.2 * d_y_r_active_inf_area [l]) / 2
d_M_b_l_y_active_inf_area [l] = 1.2 * d_y_l_active_inf_area [l] + (1.2 *
d_y_b_active_inf_area [l] - 1.2 * d_y_l_active_inf_area [l]) / 2
d_M_b_r_y_active_inf_area [l] = 1.2 * d_y_r_active_inf_area [l] + (1.2 *
d_y_b_active_inf_area [l] - 1.2 * d_y_r_active_inf_area [l]) / 2
}

# Set active infection area of last point (escape point) equal to point before
d_x_f_active_inf_area [length(x_agent_new)] = d_x_f_active_inf_area
[length(x_agent_new)-1]
d_y_f_active_inf_area [length(x_agent_new)] = d_y_f_active_inf_area
[length(x_agent_new)-1]
d_x_b_active_inf_area [length(x_agent_new)] = d_x_b_active_inf_area
[length(x_agent_new)-1]
d_y_b_active_inf_area [length(x_agent_new)] = d_y_b_active_inf_area
[length(x_agent_new)-1]
d_x_r_active_inf_area [length(x_agent_new)] = d_x_r_active_inf_area
[length(x_agent_new)-1]
d_y_r_active_inf_area [length(x_agent_new)] = d_y_r_active_inf_area
[length(x_agent_new)-1]
d_x_l_active_inf_area [length(x_agent_new)] = d_x_l_active_inf_area
[length(x_agent_new)-1]
d_y_l_active_inf_area [length(x_agent_new)] = d_y_l_active_inf_area
[length(x_agent_new)-1]

d_M_f_r_x_active_inf_area [length(x_agent_new)] = d_M_f_r_x_active_inf_area
[length(x_agent_new)-1]
d_M_f_r_y_active_inf_area [length(x_agent_new)] = d_M_f_r_y_active_inf_area
[length(x_agent_new)-1]
d_M_f_l_x_active_inf_area [length(x_agent_new)] = d_M_f_l_x_active_inf_area
[length(x_agent_new)-1]
```

```
d_M_f_l_y_active_inf_area [length(x_agent_new)] = d_M_f_l_y_active_inf_area
[length(x_agent_new)-1]
d_M_b_r_x_active_inf_area [length(x_agent_new)] = d_M_b_r_x_active_inf_area
[length(x_agent_new)-1]
d_M_b_r_y_active_inf_area [length(x_agent_new)] = d_M_b_r_y_active_inf_area
[length(x_agent_new)-1]
d_M_b_l_x_active_inf_area [length(x_agent_new)] = d_M_b_l_x_active_inf_area
[length(x_agent_new)-1]
d_M_b_l_y_active_inf_area [length(x_agent_new)] = d_M_b_l_y_active_inf_area
[length(x_agent_new)-1]
}

# For each agent - Combine parameters in one table
# Include agent profile
agent_profile = array(profiles_info$profile [profiles_info$id == j],
length(time_agent_new))
agent_status  = array(status_agent, length(time_agent_new))
agent_masked  = array(masked_agent, length(time_agent_new))

agent_info_new = data.frame (
 id_agent_new, agent_profile, agent_status, agent_masked, time_agent_new,
x_agent_new, y_agent_new, v_agent_new,
 d_x_f_active_inf_area, d_y_f_active_inf_area,
 d_M_f_r_x_active_inf_area, d_M_f_r_y_active_inf_area,
 d_x_r_active_inf_area, d_y_r_active_inf_area,
 d_M_b_r_x_active_inf_area, d_M_b_r_y_active_inf_area,
 d_x_b_active_inf_area, d_y_b_active_inf_area,
 d_M_b_l_x_active_inf_area, d_M_b_l_y_active_inf_area,
 d_x_l_active_inf_area, d_y_l_active_inf_area,
 d_M_f_l_x_active_inf_area, d_M_f_l_y_active_inf_area
)
colnames(agent_info_new) = c(
 "id", "profile", "status", "masked", "time", "x", "y", "v",
 "d_x_f", "d_y_f", "d_M_f_r_x", "d_M_f_r_y",
 "d_x_r", "d_y_r", "d_M_b_r_x", "d_M_b_r_y",
 "d_x_b", "d_y_b", "d_M_b_l_x", "d_M_b_l_y",
 "d_x_l", "d_y_l", "d_M_f_l_x", "d_M_f_l_y"
)
# Add individual parameters to one table
output_table = rbind (output_table, agent_info_new)
}
# Calculate edge points for different infection areas
# Intermediate and casual contact distances based on expansion parameter e com-
pared to casual contact
# Revisit values!
e_interm = 1.5
e_casual = 2.0

# casual contact area

x_f_close= output_table$x + output_table$d_x_f
y_f_close= output_table$y + output_table$d_y_f
x_b_close= output_table$x + output_table$d_x_b
y_b_close= output_table$y + output_table$d_y_b
x_r_close= output_table$x + output_table$d_x_r
y_r_close= output_table$y + output_table$d_y_r
x_l_close= output_table$x + output_table$d_x_l
y_l_close= output_table$y + output_table$d_y_l

M_f_r_x_close= output_table$x + output_table$d_M_f_r_x
M_f_r_y_close = output_table$y + output_table$d_M_f_r_y
M_f_l_x_close = output_table$x + output_table$d_M_f_l_x
M_f_l_y_close = output_table$y + output_table$d_M_f_l_y
M_b_r_x_close = output_table$x + output_table$d_M_b_r_x
M_b_r_y_close = output_table$y + output_table$d_M_b_r_y
M_b_l_x_close = output_table$x + output_table$d_M_b_l_x
M_b_l_y_close = output_table$y + output_table$d_M_b_l_y

# Intermediate contact area
```

```
x_f_interm= output_table$x + e_interm * output_table$d_x_f
y_f_interm= output_table$y + e_interm * output_table$d_y_f
x_b_interm= output_table$x + e_interm * output_table$d_x_b
y_b_interm= output_table$y + e_interm * output_table$d_y_b
x_r_interm= output_table$x + e_interm * output_table$d_x_r
y_r_interm= output_table$y + e_interm * output_table$d_y_r
x_l_interm= output_table$x + e_interm * output_table$d_x_l
y_l_interm= output_table$y + e_interm * output_table$d_y_l

M_f_r_x_interm= output_table$x + e_interm * output_table$d_M_f_r_x
M_f_r_y_interm= output_table$y + e_interm * output_table$d_M_f_r_y
M_f_l_x_interm= output_table$x + e_interm * output_table$d_M_f_l_x
M_f_l_y_interm= output_table$y + e_interm * output_table$d_M_f_l_y
M_b_r_x_interm= output_table$x + e_interm * output_table$d_M_b_r_x
M_b_r_y_interm= output_table$y + e_interm * output_table$d_M_b_r_y
M_b_l_x_interm= output_table$x + e_interm * output_table$d_M_b_l_x
M_b_l_y_interm= output_table$y + e_interm * output_table$d_M_b_l_y

# Casual contact area

x_f_casual= output_table$x + e_casual * output_table$d_x_f
y_f_casual= output_table$y + e_casual * output_table$d_y_f
x_b_casual= output_table$x + e_casual * output_table$d_x_b
y_b_casual= output_table$y + e_casual * output_table$d_y_b
x_r_casual= output_table$x + e_casual * output_table$d_x_r
y_r_casual= output_table$y + e_casual * output_table$d_y_r
x_l_casual= output_table$x + e_casual * output_table$d_x_l
y_l_casual= output_table$y + e_casual * output_table$d_y_l

M_f_r_x_casual= output_table$x + e_casual * output_table$d_M_f_r_x
M_f_r_y_casual= output_table$y + e_casual * output_table$d_M_f_r_y
M_f_l_x_casual= output_table$x + e_casual * output_table$d_M_f_l_x
M_f_l_y_casual= output_table$y + e_casual * output_table$d_M_f_l_y
M_b_r_x_casual= output_table$x + e_casual * output_table$d_M_b_r_x
M_b_r_y_casual= output_table$y + e_casual * output_table$d_M_b_r_y
M_b_l_x_casual= output_table$x + e_casual * output_table$d_M_b_l_x
M_b_l_y_casual= output_table$y + e_casual * output_table$d_M_b_l_y

# Delete not needed distances
output_table = output_table[-c(9:24)]

# Add edge points infection area
output_table = cbind (output_table,
x_f_close, y_f_close, x_b_close, y_b_close, x_r_close, y_r_close, x_l_close,
y_l_close,
M_f_r_x_close, M_f_r_y_close, M_f_l_x_close, M_f_l_y_close, M_b_r_x_close,
M_b_r_y_close, M_b_l_x_close, M_b_l_y_close,
x_f_interm, y_f_interm, x_b_interm, y_b_interm, x_r_interm, y_r_interm, x_l_in-
term, y_l_interm,
M_f_r_x_interm, M_f_r_y_interm, M_f_l_x_interm, M_f_l_y_interm, M_b_r_x_interm,
M_b_r_y_interm, M_b_l_x_interm, M_b_l_y_interm,
x_f_casual, y_f_casual, x_b_casual, y_b_casual, x_r_casual, y_r_casual, x_l_cas-
ual, y_l_casual,
M_f_r_x_casual, M_f_r_y_casual, M_f_l_x_casual, M_f_l_y_casual, M_b_r_x_casual,
M_b_r_y_casual, M_b_l_x_casual, M_b_l_y_casual
)

# Show results (control)
output_table[1:10,]

# Export results
export = 0 # export if 1, do nothing if 0

if (export == 1) {
name_output_file = "Revised dataset.txt"
path_output_file = path_input_file
output_file      = paste(path_output_file, name_output_file, sep="" )
write.table (output_table, file = output_file, row.names = FALSE, col.names =
TRUE, sep = "\t")
}
```

12.3.3 Script 2 – Crossings infection areas

```
## remove variables in the main environment
rm(list=ls())

# Import input from A1 output
name_of_file        = "Revised dataset.txt"
path_input_file     = paste(getwd(),"/",sep="")
input_file          = paste(path_input_file ,name_of_file, sep="" )
agents_info_revised = read.table(file = input_file, header = TRUE)

# Import agent profile parameters from A0
name_of_profiles_file   = "Agent parameters.txt"
input_file_profiles     = paste(path_input_file, name_of_profiles_file, sep="" )
profiles_info = read.table(file = input_file_profiles, skip=0, header = TRUE)

# Re-assign variables -> better overview
id= agents_info_revised$id
time= agents_info_revised$time
x= agents_info_revised$x
y= agents_info_revised$y
v= agents_info_revised$v

# Assign factor reduced transmission probability for vaccinated agents
normal_transmission = 1
vacc_transmission   = 0.5

# Calculate infection areas
# Define funtion and variables
area_function <- function(xvec, yvec){
s1 = 0
s2 = 0
for (i in 1:(length(xvec)-1)){
  s1 = (yvec[i]*xvec[i+1]) + s1
  s2 = (xvec[i]*yvec[i+1]) + s2
}
area_function = 0.5*(s1-s2)
}

# Define close, intermediate, and casual contact areas
a_infec_close  = array (0, length(id))
a_infec_interm = array (0, length(id))
a_infec_casual = array (0, length(id))

contacts   = array (0, length(id))
temp_x_vec = array (0, 9)
temp_y_vec = array (0, 9)

# Create output table
output_table = data.frame (agents_info_revised, a_infec_close, a_infec_interm,
a_infec_casual, contacts)

# Divided in infected, susceptible, vaccinated, and recovered
output_infected    = subset(output_table, status == "I" | status == "VI")
output_susceptible = subset(output_table, status == "S" | status == "V")
output_recovered   = subset(output_table, status == "R" )

# infection areas just for infected agents
# Close contact
for (k in 1:(length(output_infected$id))) {
temp_x_vec = c(output_infected$x_f_close[k]    , output_in-
fected$M_f_r_x_close[k], output_infected$x_r_close [k]   , output_in-
fected$M_b_r_x_close[k], output_infected$x_b_close[k],
   output_infected$M_b_l_x_close[k], output_infected$x_l_close[k]    , output_in-
fected$M_f_l_x_close[k], output_infected$x_f_close[k])
temp_y_vec = c(output_infected$y_f_close[k]    , output_in-
fected$M_f_r_y_close[k], output_infected$y_r_close[k]    , output_in-
fected$M_b_r_y_close[k], output_infected$y_b_close[k],
   output_infected$M_b_l_y_close[k], output_infected$y_l_close[k]    , output_in-
fected$M_f_l_y_close[k], output_infected$y_f_close[k])
```

```
output_infected$a_infec_close[k] = abs(area_function(temp_x_vec, temp_y_vec))

# Intermediate contact
temp_x_vec = c(output_infected$x_f_interm[k]    , output_infected$M_f_r_x_in-
term[k], output_infected$x_r_interm [k]   , output_infected$M_b_r_x_interm[k],
output_infected$x_b_interm[k],
   output_infected$M_b_l_x_interm[k], output_infected$x_l_interm[k]    , out-
put_infected$M_f_l_x_interm[k], output_infected$x_f_interm[k])
temp_y_vec = c(output_infected$y_f_interm[k]    , output_infected$M_f_r_y_in-
term[k], output_infected$y_r_interm[k]    , output_infected$M_b_r_y_interm[k],
output_infected$y_b_interm[k],
   output_infected$M_b_l_y_interm[k], output_infected$y_l_interm[k]    , out-
put_infected$M_f_l_y_interm[k], output_infected$y_f_interm[k])
output_infected$a_infec_interm[k] = abs(area_function(temp_x_vec, temp_y_vec))

# Casual contact
temp_x_vec = c(output_infected$x_f_casual[k]    , output_infected$M_f_r_x_cas-
ual[k], output_infected$x_r_casual [k]   , output_infected$M_b_r_x_casual[k],
output_infected$x_b_casual[k],
   output_infected$M_b_l_x_casual[k], output_infected$x_l_casual[k]    , out-
put_infected$M_f_l_x_casual[k], output_infected$x_f_casual[k])
temp_y_vec = c(output_infected$y_f_casual[k]    , output_infected$M_f_r_y_cas-
ual[k], output_infected$y_r_casual[k]    , output_infected$M_b_r_y_casual[k],
output_infected$y_b_casual[k],
   output_infected$M_b_l_y_casual[k], output_infected$y_l_casual[k]    , out-
put_infected$M_f_l_y_casual[k], output_infected$y_f_casual[k])
output_infected$a_infec_casual[k] = abs(area_function(temp_x_vec, temp_y_vec))
}

# Check if uninfected agents cross infection zones
for (l in 1:(length(output_susceptible$id))) {
for (k in 1:(length(output_infected$id))) {

a_cont = 0

# Include less transmission by vaccinated agents
if (output_infected$status [k] == "VI") {
transmission = vacc_transmission
}
else {transmission = normal_transmission}

# Check for casual contact

#1
temp_x_vec = c(output_infected$x_f_casual[k], output_infected$M_f_r_x_casual[k],
output_susceptible$x [l], output_infected$x_f_casual[k] )
temp_y_vec = c(output_infected$y_f_casual[k], output_infected$M_f_r_y_casual[k],
output_susceptible$y [l], output_infected$y_f_casual[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#2
temp_x_vec = c(output_infected$M_f_r_x_casual[k], output_infected$x_r_casual[k],
output_susceptible$x[l], output_infected$M_f_r_x_casual[k] )
temp_y_vec = c(output_infected$M_f_r_y_casual[k], output_infected$y_r_casual[k],
output_susceptible$y[l], output_infected$M_f_r_y_casual[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#3
temp_x_vec = c(output_infected$x_r_casual[k], output_infected$M_b_r_x_casual[k],
output_susceptible$x[l], output_infected$x_r_casual[k] )
temp_y_vec = c(output_infected$y_r_casual[k], output_infected$M_b_r_y_casual[k],
output_susceptible$y[l], output_infected$y_r_casual[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#4
temp_x_vec = c(output_infected$M_b_r_x_casual[k], output_infected$x_b_casual[k],
output_susceptible$x[l], output_infected$M_b_r_x_casual[k] )
temp_y_vec = c(output_infected$M_b_r_y_casual[k], output_infected$y_b_casual[k],
output_susceptible$y[l], output_infected$M_b_r_y_casual[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))
```

```
#5
temp_x_vec = c(output_infected$x_b_casual[k], output_infected$M_b_l_x_casual[k],
output_susceptible$x[l], output_infected$x_b_casual[k] )
temp_y_vec = c(output_infected$y_b_casual[k], output_infected$M_b_l_y_casual[k],
output_susceptible$y[l], output_infected$y_b_casual[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#6
temp_x_vec = c(output_infected$M_b_l_x_casual[k], output_infected$x_l_casual[k],
output_susceptible$x[l], output_infected$M_b_l_x_casual[k] )
temp_y_vec = c(output_infected$M_b_l_y_casual[k], output_infected$y_l_casual[k],
output_susceptible$y[l], output_infected$M_b_l_y_casual[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#7
temp_x_vec = c(output_infected$x_l_casual[k], output_infected$M_f_l_x_casual[k],
output_susceptible$x[l], output_infected$x_l_casual[k] )
temp_y_vec = c(output_infected$y_l_casual[k], output_infected$M_f_l_y_casual[k],
output_susceptible$y[l], output_infected$y_l_casual[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#8
temp_x_vec = c(output_infected$M_f_l_x_casual[k], output_infected$x_f_casual[k],
output_susceptible$x[l], output_infected$M_f_l_x_casual[k] )
temp_y_vec = c(output_infected$M_f_l_y_casual[k], output_infected$y_f_casual[k],
output_susceptible$y[l], output_infected$M_f_l_y_casual[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

if (a_cont == output_infected$a_infec_casual[k]) {
if ((0 <= output_susceptible$time [l] - output_infected$time [k]) &
     (    output_susceptible$time [l] - output_infected$time [k]) <= 0.25) {
output_susceptible$contacts [l] = output_susceptible$contacts [l] + transmission
* 0.25

# Check for intermediate contact

a_cont = 0

#1
temp_x_vec = c(output_infected$x_f_interm[k], output_infected$M_f_r_x_interm[k],
output_susceptible$x [l], output_infected$x_f_interm[k] )
temp_y_vec = c(output_infected$y_f_interm[k], output_infected$M_f_r_y_interm[k],
output_susceptible$y [l], output_infected$y_f_interm[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#2
temp_x_vec = c(output_infected$M_f_r_x_interm[k], output_infected$x_r_interm[k],
output_susceptible$x[l], output_infected$M_f_r_x_interm[k] )
temp_y_vec = c(output_infected$M_f_r_y_interm[k], output_infected$y_r_interm[k],
output_susceptible$y[l], output_infected$M_f_r_y_interm[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#3
temp_x_vec = c(output_infected$x_r_interm[k], output_infected$M_b_r_x_interm[k],
output_susceptible$x[l], output_infected$x_r_interm[k] )
temp_y_vec = c(output_infected$y_r_interm[k], output_infected$M_b_r_y_interm[k],
output_susceptible$y[l], output_infected$y_r_interm[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#4
temp_x_vec = c(output_infected$M_b_r_x_interm[k], output_infected$x_b_interm[k],
output_susceptible$x[l], output_infected$M_b_r_x_interm[k] )
temp_y_vec = c(output_infected$M_b_r_y_interm[k], output_infected$y_b_interm[k],
output_susceptible$y[l], output_infected$M_b_r_y_interm[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#5
temp_x_vec = c(output_infected$x_b_interm[k], output_infected$M_b_l_x_interm[k],
output_susceptible$x[l], output_infected$x_b_interm[k] )
```

```
temp_y_vec = c(output_infected$y_b_interm[k], output_infected$M_b_l_y_interm[k],
output_susceptible$y[l], output_infected$y_b_interm[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#6
temp_x_vec = c(output_infected$M_b_l_x_interm[k], output_infected$x_l_interm[k],
output_susceptible$x[l], output_infected$M_b_l_x_interm[k] )
temp_y_vec = c(output_infected$M_b_l_y_interm[k], output_infected$y_l_interm[k],
output_susceptible$y[l], output_infected$M_b_l_y_interm[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#7
temp_x_vec = c(output_infected$x_l_interm[k], output_infected$M_f_l_x_interm[k],
output_susceptible$x[l], output_infected$x_l_interm[k] )
temp_y_vec = c(output_infected$y_l_interm[k], output_infected$M_f_l_y_interm[k],
output_susceptible$y[l], output_infected$y_l_interm[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#8
temp_x_vec = c(output_infected$M_f_l_x_interm[k], output_infected$x_f_interm[k],
output_susceptible$x[l], output_infected$M_f_l_x_interm[k] )
temp_y_vec = c(output_infected$M_f_l_y_interm[k], output_infected$y_f_interm[k],
output_susceptible$y[l], output_infected$M_f_l_y_interm[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

if (a_cont == output_infected$a_infec_interm[k]) {
if ((0 <= output_susceptible$time [l] - output_infected$time [k]) &
(     output_susceptible$time [l] - output_infected$time [k]) <= 0.25) {
output_susceptible$contacts [l] = output_susceptible$contacts [l] + transmission
* 0.25

# Check for close contact
a_cont = 0

#1
temp_x_vec = c(output_infected$x_f_close[k], output_infected$M_f_r_x_close[k],
output_susceptible$x [l], output_infected$x_f_close[k] )
temp_y_vec = c(output_infected$y_f_close[k], output_infected$M_f_r_y_close[k],
output_susceptible$y [l], output_infected$y_f_close[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#2
temp_x_vec = c(output_infected$M_f_r_x_close[k], output_infected$x_r_close[k],
output_susceptible$x[l], output_infected$M_f_r_x_close[k] )
temp_y_vec = c(output_infected$M_f_r_y_close[k], output_infected$y_r_close[k],
output_susceptible$y[l], output_infected$M_f_r_y_close[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#3
temp_x_vec = c(output_infected$x_r_close[k], output_infected$M_b_r_x_close[k],
output_susceptible$x[l], output_infected$x_r_close[k] )
temp_y_vec = c(output_infected$y_r_close[k], output_infected$M_b_r_y_close[k],
output_susceptible$y[l], output_infected$y_r_close[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#4
temp_x_vec = c(output_infected$M_b_r_x_close[k], output_infected$x_b_close[k],
output_susceptible$x[l], output_infected$M_b_r_x_close[k] )
temp_y_vec = c(output_infected$M_b_r_y_close[k], output_infected$y_b_close[k],
output_susceptible$y[l], output_infected$M_b_r_y_close[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#5
temp_x_vec = c(output_infected$x_b_close[k], output_infected$M_b_l_x_close[k],
output_susceptible$x[l], output_infected$x_b_close[k] )
temp_y_vec = c(output_infected$y_b_close[k], output_infected$M_b_l_y_close[k],
output_susceptible$y[l], output_infected$y_b_close[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#6
```

```
temp_x_vec = c(output_infected$M_b_l_x_close[k], output_infected$x_l_close[k],
output_susceptible$x[l], output_infected$M_b_l_x_close[k] )
temp_y_vec = c(output_infected$M_b_l_y_close[k], output_infected$y_l_close[k],
output_susceptible$y[l], output_infected$M_b_l_y_close[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#7
temp_x_vec = c(output_infected$x_l_close[k], output_infected$M_f_l_x_close[k],
output_susceptible$x[l], output_infected$x_l_close[k] )
temp_y_vec = c(output_infected$y_l_close[k], output_infected$M_f_l_y_close[k],
output_susceptible$y[l], output_infected$y_l_close[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

#8
temp_x_vec = c(output_infected$M_f_l_x_close[k], output_infected$x_f_close[k],
output_susceptible$x[l], output_infected$M_f_l_x_close[k] )
temp_y_vec = c(output_infected$M_f_l_y_close[k], output_infected$y_f_close[k],
output_susceptible$y[l], output_infected$M_f_l_y_close[k] )
a_cont = a_cont + abs(area_function(temp_x_vec, temp_y_vec))

if (a_cont == output_infected$a_infec_close[k]) {
if ((0 <= output_susceptible$time [l] - output_infected$time [k]) &
(    output_susceptible$time [l] - output_infected$time [k]) <= 0.25) {
output_susceptible$contacts [l] = output_susceptible$contacts [l] + transmission
* 0.5
}
}
}
}
}
}
}
}

# Re-combine output_table
output_table = rbind (output_infected, output_susceptible, output_recovered)

# Shortened result tables for export
result_contacts_id  = array(0, (max(id))+1)
start_time = array(0, (max(id))+1)
end_time = array(0, (max(id))+1)
t_exp = array(0, (max(id))+1)

for (m in 0:(max(id))) {
result_contacts_id [m+1] = sum (output_table$contacts [output_table$id == m])
start_time [m+1]  = min (output_table$time [output_table$id == m])
end_time [m+1]  = max (output_table$time [output_table$id == m])
t_exp [m+1] = end_time [m+1] - start_time [m+1]
}

result_table = cbind (profiles_info [,1:2], profiles_info [4:5], start_time,
end_time, t_exp, result_contacts_id)
colnames(result_table) = c("id", "profile", "start_status", "masked",
"start_time", "end_time", "exposure_time", "total_contacts")
result_table

# Export results
export = 1 # export if 1, do nothing if 0

if (export == 1) {
name_output_file_1    = "All geometric results.txt"
name_output_file_2    = "Basic geometric results.txt"
path_output_file   = path_input_file
output_file_1         = paste(path_output_file, name_output_file_1, sep="" )
output_file_2         = paste(path_output_file, name_output_file_2, sep="" )
write.table (output_table, file = output_file_1, row.names = FALSE, col.names =
TRUE, sep = "\t")
write.table (result_table, file = output_file_2, row.names = FALSE, col.names =
TRUE, sep = "\t")
}
```

12.3.4 Script 3 – Probability calculations

```
## remove variables in the main environment
rm(list=ls())

# Import input from A2 output
name_of_file          = "Basic geometric results.txt"
path_input_file     = paste(getwd(),"/",sep="")
input_file            = paste(path_input_file ,name_of_file, sep="" )
contacts_info_revised = read.table(file = input_file, header = TRUE)

# Create overview variables

number_agents   = length(contacts_info_revised$id)
number_infected_start = length(contacts_info_revised$id[contacts_info_re-
vised$start_status == "I" | contacts_info_revised$start_status == "VI"])
number_susceptible   = length(contacts_info_revised$id[contacts_info_re-
vised$start_status == "S"])
number_new_infected   = 0

model_area = 5 * 5 # m²
h_room = 3 # m

# Create vectors for infection probabilities, end status of agents and highlight
of new infected
p_active_infec  = array(0, number_agents)
p_passive_infec = array(0, number_agents)
p_infec_total= array(0, number_agents)

# Reduced infection probability considering vaccination and face masks
p_infec_total_red = array(0, number_agents)

end_status   = as.character(contacts_info_revised$start_status)
new_infected = array(0, number_agents)

# Assign mask filter and vaccine protection percentage
mask_filter_perc = 80
vacc_protection_perc = 50

# Assign infection probabilities
# Passive infection depending on room characteristics
# Base passive infection probability based on concentration of infected agents
[1/m²] and room height [m]
c_inf_agents = number_infected_start / model_area

p_passive_base = c_inf_agents * 0.5 / h_room

# Influence coefficients for air exchange rate [1/h]
r_ae = 2 # = 2 * room volume per hour

if   (r_ae <  1) {C_ae = 1.0
} else if (r_ae <= 3) {C_ae = 0.8
} else if (r_ae <= 6) {C_ae = 0.6
} else   {C_ae = 0.5}

# Active infection depending on number of contacts
for (i in 1:number_agents) {
if (contacts_info_revised$start_status [i] == "S" | contacts_info_re-
vised$start_status [i] == "V") {

# S agent not wearing a mask nor being vaccinated (base case)
# Active infection probability
if   (contacts_info_revised$total_contacts [i] == 0) {p_active_infec [i] =
0.0000001
} else if (contacts_info_revised$total_contacts [i] <=  3) {p_active_infec [i] =
0.25
} else if (contacts_info_revised$total_contacts [i] <=  8) {p_active_infec [i] =
0.50
} else if (contacts_info_revised$total_contacts [i] <= 15){p_active_infec [i] =
0.75
```

```
} else {p_active_infec [i] = 0.95}

# Passive infection probability
# Influence of the exposure time [min]
if (contacts_info_revised$exposure_time [i] / 60 <  1) {C_t_exp = 0.5}
else if (contacts_info_revised$exposure_time [i] / 60 <= 3) {C_t_exp = 0.6}
else if (contacts_info_revised$exposure_time [i] / 60 <= 6) {C_t_exp = 0.8}
else {C_t_exp = 1.0}

p_passive_infec [i] = C_ae * C_t_exp * p_passive_base
if (p_passive_infec [i] < 0.05) {p_passive_infec [i] = 0.05}
if (p_passive_infec [i] > 0.30) {p_passive_infec [i] = 0.30}

# Resulting infection probability
p_infec_total [i] = p_active_infec [i] + p_passive_infec [i]
if (p_infec_total [i] > 1) {p_infec_total [i] = 0.99}

p_infec_total_red [i] = p_infec_total [i]

# If agent is wearing a mask
if (contacts_info_revised$masked [i] == "X") {p_infec_total_red [i] = (1 -
mask_filter_perc / 100) * p_infec_total_red [i]}

# If agent is vaccinated
if (contacts_info_revised$start_status [i] == "V") {p_infec_total_red [i] = (1 -
vacc_protection_perc / 100) * p_infec_total_red [i]}

# Calculate infection outcome based on probability assinged above
control_val = sample(c(0,1), size = 1, replace=FALSE, prob=(c(1 - p_infec_to-
tal_red [i], p_infec_total_red [i] )))

if (contacts_info_revised$start_status [i] == "S") {
if (control_val == 0 )  {end_status[i] = "S"}
else {end_status [i] = "I"; number_new_infected = number_new_infected + 1}
}

if (contacts_info_revised$start_status [i] == "V") {
if (control_val == 0 )  {end_status[i] = "V"}
else {end_status [i] = "VI"; number_new_infected = number_new_infected + 1}
}

# Highlight new infected agents
if (contacts_info_revised$start_status [i] == "S" & end_status [i] == "I" )
{new_infected [i] = "X"}
if (contacts_info_revised$start_status [i] == "V" & end_status [i] == "VI")
{new_infected [i] = "X"}
}
}

# Create final results table
final_results = cbind(contacts_info_revised, round(p_active_infec, digits = 2),
  round(p_passive_infec, digits = 2), round(p_infec_total, digits = 2),
round(p_infec_total_red, digits = 2), end_status, new_infected)

colnames(final_results) = c("id", "profile", "start_status", "masked",
"start_time", "end_time", "exposure_time", "total_contacts",
"p_active_infec", "p_passive_infec", "p_infec_total", "p_infec_total_red",
"end_status", "new_infected")

# Show results
final_results

number_agents
number_infected_start
number_susceptible
number_new_infected

# Export results
export = 0 # export if 1, do nothing if 0
```

```
if (export == 1) {
name_output_file = "Final results.txt"
path_output_file = path_input_file
output_file      = paste(path_output_file, name_output_file, sep="" )
write.table (final_results, file = output_file, row.names = FALSE, col.names =
TRUE, sep = "\t")
}
```

12.3.5 Automation of simulation repetitions

```
######################## LIBRARIES
# Library for python - automation of GUI (Graphic User Interface)
import pyautogui
# Library for controlling time
import time
# Libraries for controlling OS (operating System) processes
import os
import shutil
import subprocess
import sys

###################### INPUT
### Paths
# Pathfinder info
pathfinder_exe = "C:\\Program Files\\Pathfinder 2021\\pathfinder.exe"
# Source info
folder_source_files =  'C:\\Users\mirco\Aalborg Universitet\\2021 - MH - Disease
Transmission in Public Spaces - General\\Simulations_final\\Complex model\\with
obstacles\\Source files\\'
source_file      =  'complext with obstacles.pth'
source_R_0    =  'C_WO_A0_03_MH_Script_agent profile parameters.r'
source_R_1    =  'C_WO_A1_08_MH_Script_Interpolation + Infection area.r'
source_R_2    =  'C_WO_A2_04_MH_Script_Crossings infection area.r'
source_R_3    =  'C_WO_A3_02_MH_Script_Probability calculations.r'

final_source_file   =  folder_source_files + source_file
final_R_0           =  folder_source_files + source_R_0
final_R_1           =  folder_source_files + source_R_1
final_R_2           =  folder_source_files + source_R_2
final_R_3           =  folder_source_files + source_R_3

# Destination info
simulation_folder   =  'C:\\Users\mirco\Aalborg Universitet\\2021 - MH - Disease
Transmission in Public Spaces - General\\Simulations_final\\Complex model\\with
obstacles\\'
generic_file_name   =  'complext with obstacles'
generic_folder_name =  'SIM'

### Other input
nsim    = 20
computer_stress_factor =  1.3

######### Create folder and later Copy-paste the source file
access_rights = 0o755 # define the access rights

## Create again the simulation folder for each case of simulation.
try:
   os.mkdir(simulation_folder, access_rights)
except OSError:
   print("Creation of the directory %s failed")
else:
   print("Creation of the directory %s succeded")


### Create folder and copy-paste source file
for i in range(1, nsim):
   # Print simulation number started
   print("Simulation number " + str(i) + " started")

   subfolder_path = simulation_folder + generic_folder_name + '_'  +
str('{:04d}'.format(i) )
   try:
      os.makedirs(subfolder_path, access_rights)
   except OSError:
      print('Creation of the subdirectory %s failed' % subfolder_path)
   else:
      print('Successfully created the subdirectory %s' % subfolder_path )
```

```
   destiny_folder_file = subfolder_path + '\\'  + generic_file_name + '.pth'
   destiny_R_0 = subfolder_path + '\\'  + source_R_0
   destiny_R_1 = subfolder_path + '\\'  + source_R_1
   destiny_R_2 = subfolder_path + '\\'  + source_R_2
   destiny_R_3 = subfolder_path + '\\'  + source_R_3

   print (destiny_folder_file)
   shutil.copyfile(final_source_file, destiny_folder_file)
   shutil.copyfile(final_R_0, destiny_R_0)
   shutil.copyfile(final_R_1, destiny_R_1)
   shutil.copyfile(final_R_2, destiny_R_2)
   shutil.copyfile(final_R_3, destiny_R_3)

   ###### Python automation - open pathfinder
   P = subprocess.Popen([ pathfinder_exe   ,  destiny_folder_file   ])
   time.sleep(10*computer_stress_factor)

   # Randomise agent parameters
   # Open context menu
   pyautogui.click(x = 132, y = 507, clicks=1, interval = 0.0, button='right',
duration=1.0)
   time.sleep(2*computer_stress_factor)
   # Randomise
   pyautogui.click(x = 272, y = 826, clicks=1, interval = 0.0, button='left', du-
ration=1.0)
   time.sleep(3*computer_stress_factor)

   # Randomise agent location
   # Open context menu
   pyautogui.click(x = 106, y = 671, clicks=1, interval = 0.0, button='right',
duration=1.0)
   time.sleep(2*computer_stress_factor)
   # Randomise
   pyautogui.click(x = 236, y = 1284, clicks=1, interval = 0.0, button='left',
duration=1.0)
   time.sleep(2*computer_stress_factor)
   # Confirm
   pyautogui.press('enter')
   time.sleep(3*computer_stress_factor)

   # Only for movement group scenario
   # Create movement groups
   # Open context menu
   pyautogui.click(x = 116, y = 510, clicks=1, interval = 0.0, button='right',
duration=1.0)
   time.sleep(2*computer_stress_factor)
   # Create movement group from template
   pyautogui.click(x = 342, y = 970, clicks=1, interval = 0.0, button='left', du-
ration=1.0)
   time.sleep(2*computer_stress_factor)
   # Open distribution window
   pyautogui.click(x = 1198, y = 766, clicks=1, interval = 0.0, button='left',
duration=1.0)
   time.sleep(2*computer_stress_factor)
   # Select first row
   pyautogui.click(x = 1228, y = 760, clicks=1, interval = 0.0, button='left',
duration=1.0)
   time.sleep(2*computer_stress_factor)
   # Type in percentages row by row
   pyautogui.typewrite('30') # Double
   pyautogui.press('enter')
   pyautogui.typewrite('10') # Quartet
   pyautogui.press('enter')
   pyautogui.typewrite('40') # Single
   pyautogui.press('enter')
   pyautogui.typewrite('20') # Triple
   pyautogui.press('enter')
   # Confirm distribution window
   pyautogui.click(x = 1336, y = 1098, clicks=1, interval = 0.0, button='left',
duration=1.0)
```

```
   time.sleep(2*computer_stress_factor)
   # Confirm create movement group
   pyautogui.click(x = 1390, y = 1075, clicks=1, interval = 0.0, button='left',
duration=1.0)
   time.sleep(2*computer_stress_factor)
   # Collapse all expended categories
   pyautogui.click(x = 72, y = 238, clicks=1, interval = 0.0, button='left', du-
ration=1.0)
   time.sleep(2*computer_stress_factor)

   # Save
   pyautogui.hotkey('ctrl', 's')
   time.sleep(4*computer_stress_factor)

   # Run simulation
   pyautogui.hotkey('alt', 's') # Simulation in main menu
   pyautogui.press('down')
   pyautogui.press('enter')
   time.sleep(20*computer_stress_factor)
   pyautogui.click(x = 1747, y=1242, clicks=1, interval = 0.0, button='left', du-
ration=1.0)  # ok after running

   # Save
   pyautogui.hotkey('ctrl', 's')
   time.sleep(3*computer_stress_factor)

   ## Close the file
   pyautogui.hotkey('alt', 'f')  # File in main menu
   pyautogui.press('up')
   pyautogui.press('enter')
   time.sleep(3*computer_stress_factor)

   # Set active directory to subfolder_path
   os.chdir(subfolder_path)

   ### Run R
   R = 'C:/Program Files/R/R-3.6.2/bin/Rscript'
   variant = '--vanilla'

   #### R_0
   run_R_0 = destiny_R_0.replace("\\","/")
   pr_0 = subprocess.Popen([R, variant, run_R_0], shell=True)
   pr_0.wait()
   time.sleep(5*computer_stress_factor)

   #### R_1
   run_R_1 = destiny_R_1.replace("\\","/")
   pr_1 = subprocess.Popen([R, variant, run_R_1], shell=True)
   pr_1.wait()
   time.sleep(5*computer_stress_factor)

   #### R_2
   run_R_2 = destiny_R_2.replace("\\","/")
   pr_2 = subprocess.Popen([R, variant, run_R_2], shell=True)
   pr_2.wait()
   time.sleep(5*computer_stress_factor)

   #### R_3
   run_R_3 = destiny_R_3.replace("\\","/")
   pr_3 = subprocess.Popen([R, variant, run_R_3], shell=True)
   pr_3.wait()

   # Print simulation number done
   print("Simulation number " + str(i) + " finished")

# Print finished message
print("XXXXXXXXXXXXXXXXXXX SIMULATIONS DONE XXXXXXXXXXXXXXXXXXX")
```

12.3.6 Result evaluation

```
## remove variables in the main environment
rm(list=ls())

# Create empty output table for outcome and complete results
outcome_table= data.frame()
complete_table = data.frame()

# Import all result files step by step and add to output table
name_of_results_file  = "Final results.txt"
scenario_path= paste(getwd(), "/", sep="")
common_path_input_file= paste(scenario_path, "SIM_00",sep="")
number_repetitions = 20

for (i in 0:(number_repetitions - 1)) {
if (i < 10) {end_path_input_file = paste("0", i, "/", sep="")}
else {end_path_input_file = paste(i, "/", sep="")}
input_file_results  = paste(common_path_input_file, end_path_input_file,
name_of_results_file, sep="" )
results_repetition = read.table(file = input_file_results, skip=0, header = TRUE)

# Agents by epidemical start status
number_agents = max(results_repetition$id) + 1
number_S_agents = dim(subset(results_repetition, start_status == "S" | start_sta-
tus == "V"))[1]
number_I_agents = dim(subset(results_repetition, start_status == "I" | start_sta-
tus == "VI"))[1]
number_R_agents = dim(subset(results_repetition, start_status == "R"))[1]
number_V_agents = dim(subset(results_repetition, start_status == "VI" |
start_status == "VS"))[1]

# Outcome evaluation
simulated_time= max(results_repetition$end_time) - min(results_repeti-
tion$start_time) # Only evaluated room
min_exposure_time= min(results_repetition$exposure_time)
mean_exposure_time= mean(results_repetition$exposure_time)
max_exposure_time= max(results_repetition$exposure_time)
total_contacts= sum(results_repetition$total_contacts)
S_without_contacts= dim(subset(results_repetition, total_contacts == 0))[1] -
(number_agents - number_S_agents)
max_contacts= max(results_repetition$total_contacts)
number_new_infected = dim(subset(results_repetition, new_infected == "X"))[1]
outcome_repetition= data.frame(repetition_number = i, simulated_time, min_expo-
sure_time, mean_exposure_time, max_exposure_time,
 total_contacts, S_without_contacts, max_contacts, number_new_infected)

# Add to outcome table
outcome_table = rbind(outcome_table, outcome_repetition)

# Add column with repetition number for complete table
repetition_number = array(i, number_agents)
results_repetition = cbind(results_repetition, repetition_number)

# Add results of single repetition to output table
complete_table = rbind(complete_table, results_repetition)
}

# Show results
outcome_table
# complete_table

# Export results
export = 1 # export if 1, do nothing if 0

if (export == 1) {
name_output_file_1    = "Summary of outcome distribution.txt"
name_output_file_2    = "Results of all repetitions.txt"
path_output_file   = scenario_path
output_file_1         = paste(path_output_file, name_output_file_1, sep="" )
```

```
output_file_2         = paste(path_output_file, name_output_file_2, sep="" )
write.table (outcome_table , file = output_file_1, row.names = FALSE, col.names =
TRUE, sep = "\t")
write.table (complete_table, file = output_file_2, row.names = FALSE, col.names =
TRUE, sep = "\t")
}
```

12.4 Result distributions

12.4.1 Minimal model

repetition number	simulated time	min exposure time	mean exposure time	max exposure time	total contacts	S without contacts	mean contacts given contact	max contacts	max p passive infec	number new infected
0	20.53	3.00	6.08	9.00	14.00	3.00	2.33	3.50	0.05	4
1	20.78	3.00	6.44	12.00	18.63	1.00	2.33	5.75	0.05	3
2	20.78	2.00	4.29	10.00	3.50	3.00	0.58	1.00	0.05	2
3	19.00	2.00	5.40	9.00	17.50	2.00	2.50	5.50	0.05	4
4	23.00	2.00	6.85	12.00	5.88	4.00	1.18	3.63	0.05	1
5	20.53	2.00	5.48	10.00	9.00	1.00	1.13	1.75	0.05	2
6	20.00	2.00	5.80	10.00	11.50	0.00	1.28	2.25	0.05	3
7	19.28	2.00	5.36	11.00	9.25	1.00	1.16	3.00	0.05	0
8	20.53	3.00	5.53	10.00	9.75	2.00	1.39	2.00	0.05	1
9	18.53	2.00	5.28	9.53	6.63	4.00	1.33	2.00	0.05	2
10	18.53	3.00	4.98	7.53	7.75	1.00	0.97	2.25	0.05	1
11	21.53	2.00	5.68	10.00	8.63	2.00	1.23	4.25	0.05	2
12	19.00	4.00	6.45	9.00	11.63	1.00	1.45	2.13	0.05	2
13	20.53	3.00	5.53	10.00	7.00	4.00	1.40	2.25	0.05	3
14	21.00	2.00	5.20	11.00	9.00	2.00	1.29	4.00	0.05	0
15	23.00	2.00	5.70	11.00	6.25	5.00	1.56	1.75	0.05	3
16	20.53	2.00	4.33	8.53	6.25	5.00	1.56	2.75	0.05	2
17	20.53	2.00	5.13	9.00	3.50	4.00	0.70	1.25	0.05	3
18	23.28	2.00	5.86	9.28	7.25	2.00	1.04	2.25	0.05	1
19	20.28	2.00	7.21	13.00	19.75	1.00	2.47	6.00	0.05	3

Table 12 - Result distribution - Minimal model

12.4.2 Public space model – Without obstacles

repetition number	simulated time	min exposure time	mean exposure time	max exposure time	total contacts	S without contacts	mean contacts given contact	max contacts	max p passive infec	number new infected
0	112.28	7.00	16.10	38.28	21.88	52.00	0.58	1.75	0.05	16
1	107.53	8.00	16.11	39.00	39.63	42.00	0.83	4.88	0.05	14
2	102.78	7.00	15.07	33.00	21.88	52.00	0.58	1.75	0.05	9
3	115.78	7.00	16.15	44.00	24.38	52.00	0.64	2.50	0.05	17
4	112.78	7.00	16.09	38.00	28.38	50.00	0.71	2.75	0.05	14
5	101.53	7.00	16.37	41.00	33.38	42.00	0.70	2.75	0.05	16
6	106.28	7.00	15.45	39.00	28.25	52.00	0.74	2.75	0.05	10
7	111.28	7.00	16.67	35.28	27.63	45.00	0.61	2.00	0.05	13
8	108.53	7.00	16.46	41.00	33.13	46.00	0.75	3.75	0.05	17
9	108.78	7.00	16.27	40.00	36.13	40.00	0.72	3.50	0.05	13
10	93.53	7.00	14.90	39.53	27.13	44.00	0.59	1.75	0.05	10
11	112.00	7.00	16.36	40.00	31.00	47.00	0.72	3.50	0.05	11
12	109.28	7.00	16.39	38.00	35.63	44.00	0.77	3.00	0.05	19
13	109.53	8.00	16.17	41.00	27.25	54.00	0.76	3.50	0.05	16
14	107.78	8.00	16.16	34.00	26.13	50.00	0.65	3.25	0.05	11
15	107.53	7.00	16.83	41.00	36.63	49.00	0.89	4.25	0.05	16
16	104.28	8.00	15.75	38.00	36.75	45.00	0.82	4.00	0.05	13
17	112.53	7.00	16.27	45.00	28.50	47.00	0.66	2.25	0.05	16
18	108.00	7.00	16.93	37.00	35.00	45.00	0.78	3.50	0.05	19
19	102.00	7.00	15.87	39.00	25.63	45.00	0.57	2.00	0.05	12

Table 13 - Result distribution - Public space model - Without obstacles

12.4.3 Public space model – With obstacles

repetition number	simulated time	min exposure time	mean exposure time	max exposure time	total contacts	S without contacts	mean contacts given contact	max contacts	max p passive infec	number new infected
0	109.00	7.00	17.90	42.00	45.25	35.00	0.82	2.00	0.05	12
1	96.53	8.00	16.32	37.00	53.13	35.00	0.97	4.50	0.05	19
2	116.78	8.00	16.68	39.78	31.88	43.00	0.68	2.75	0.05	13
3	108.00	8.00	17.52	41.00	38.25	38.00	0.74	3.00	0.05	18
4	107.28	8.00	17.36	37.00	40.13	44.00	0.87	2.75	0.05	13
5	109.53	8.00	17.27	40.00	41.38	42.00	0.86	2.50	0.05	19
6	115.00	8.00	17.17	37.00	31.63	45.00	0.70	2.25	0.05	19
7	105.78	8.00	17.17	42.00	43.25	42.00	0.90	3.75	0.05	20
8	109.00	8.00	16.21	40.00	43.88	37.00	0.83	3.00	0.05	10
9	106.53	7.00	16.53	38.00	36.00	46.00	0.82	4.25	0.05	12
10	101.78	8.00	17.03	38.00	52.38	32.00	0.90	4.75	0.05	15
11	114.28	8.00	17.53	42.00	34.88	36.00	0.65	3.25	0.05	13
12	107.78	7.00	17.23	38.00	49.25	29.00	0.81	3.75	0.05	16
13	110.28	7.00	17.17	39.00	46.13	39.00	0.90	2.75	0.05	20
14	109.28	9.00	16.52	39.00	37.63	44.00	0.82	3.75	0.05	13
15	107.53	8.00	17.24	41.00	43.38	42.00	0.90	4.00	0.05	9
16	109.28	9.00	17.44	36.00	40.38	50.00	1.01	3.50	0.05	17
17	118.78	8.00	17.67	40.00	38.50	32.00	0.66	2.13	0.05	17
18	100.53	7.00	17.34	44.00	33.63	38.00	0.65	2.25	0.05	21
19	110.28	9.00	16.88	38.00	45.88	40.00	0.92	3.25	0.05	18

Table 14 - Result distribution - Public space model - With obstacles

12.4.4 Scenario 1 – Including waiting behaviour

repetition number	simulated time	min exposure time	mean exposure time	max exposure time	total contacts	min contacts	mean contacts given contact	max contacts	max p passive infec	number new infected
0	127.78	18.00	37.23	66.00	1346.50	1.00	14.96	51.13	0.05	62
1	127.53	20.00	35.67	59.00	1442.75	0.50	16.03	55.25	0.05	60
2	119.28	16.00	36.31	63.00	1420.88	2.13	15.79	49.00	0.05	71
3	128.78	20.00	36.27	74.00	1333.00	0.75	14.81	36.88	0.05	70
4	118.00	21.00	34.59	55.00	1328.50	1.75	14.76	37.25	0.05	69
5	126.28	19.00	36.10	66.00	1416.38	0.75	15.74	57.50	0.05	62
6	120.53	19.00	36.29	63.00	1339.00	1.75	14.88	36.00	0.05	62
7	121.28	20.00	35.42	60.00	1196.50	1.25	13.29	33.50	0.05	64
8	127.53	21.00	34.42	61.00	1066.00	0.25	11.84	35.75	0.05	59
9	131.00	18.00	36.12	70.00	1110.00	0.50	12.33	32.75	0.05	66
10	124.78	19.00	35.45	64.00	1376.25	1.25	15.29	55.75	0.05	68
11	123.28	17.00	35.43	59.00	1356.75	0.75	15.08	43.50	0.05	66
12	121.28	21.00	35.86	62.00	1401.38	1.25	15.57	45.25	0.05	70
13	124.28	20.00	36.73	67.00	1392.88	1.25	15.48	42.13	0.05	57
14	131.53	20.00	38.16	69.00	1472.25	2.25	16.36	30.38	0.05	68
15	121.53	18.00	34.34	61.00	1259.50	1.38	13.99	41.63	0.05	67
16	120.28	17.00	34.66	61.00	1265.00	2.00	14.06	39.75	0.05	59
17	123.78	19.00	35.55	71.00	1174.00	1.00	13.04	40.00	0.05	63
18	117.53	19.00	33.87	58.00	1284.00	0.25	14.27	39.50	0.05	67
19	119.53	18.00	35.65	67.00	1379.50	2.00	15.33	45.25	0.05	58

Table 15 - Result distribution - Scenario 1 - Including waiting behaviour

12.4.5 Scenario 2 – Changing the demographics

repetition number	simulated time	min exposure time	mean exposure time	max exposure time	total contacts	S without contacts	mean contacts given contact	max contacts	max p passive infec	number new infected
0	100.53	7.00	16.59	39.00	24.50	48.00	0.58	2.00	0.05	12
1	103.53	9.00	17.50	41.00	20.00	59.00	0.65	1.75	0.05	15
2	112.28	7.00	16.62	43.00	24.75	58.00	0.77	3.00	0.05	11
3	108.53	8.00	16.69	40.00	23.75	52.00	0.63	2.63	0.05	12
4	103.28	7.00	16.92	40.00	27.88	48.00	0.66	3.00	0.05	18
5	110.00	9.00	17.39	41.00	26.50	53.00	0.72	6.25	0.05	12
6	112.53	9.00	17.01	38.00	23.38	46.00	0.53	2.50	0.05	17
7	108.78	8.00	17.48	41.00	19.38	63.00	0.72	2.00	0.05	14
8	113.00	8.00	17.87	38.00	27.25	52.00	0.72	4.63	0.05	12
9	103.78	8.00	17.35	45.00	25.25	46.00	0.57	1.25	0.05	16
10	104.78	7.00	17.04	40.00	26.75	51.00	0.69	2.25	0.05	10
11	111.00	9.00	17.24	40.00	24.63	53.00	0.67	2.25	0.05	17
12	108.00	8.00	16.75	36.00	22.00	49.00	0.54	2.50	0.05	13
13	107.28	8.00	17.12	40.00	27.38	50.00	0.68	3.75	0.05	17
14	100.00	9.00	17.17	41.00	25.00	59.00	0.81	3.13	0.05	11
15	107.00	7.00	16.75	39.00	25.13	56.00	0.74	2.25	0.05	10
16	95.53	7.00	16.57	41.00	20.38	62.00	0.73	3.00	0.05	7
17	108.28	9.00	17.13	47.00	28.13	53.00	0.76	3.25	0.05	14
18	108.53	7.00	17.26	41.00	28.00	55.00	0.80	3.00	0.05	8
19	110.00	9.00	17.38	43.00	19.88	55.00	0.57	2.00	0.05	9

Table 16 - Result distribution - Scenario 2 - Changing the demographics

12.4.6 Scenario 3 – Including movement groups

repetition number	simulated time	min exposure time	mean exposure time	max exposure time	total contacts	S without contacts	mean contacts given contact	max contacts	max p passive infec	number new infected
0	114.00	7.00	24.15	56.00	62.00	30.00	1.03	3.75	0.05	19
1	108.28	8.00	23.96	60.00	52.00	35.00	0.95	4.25	0.05	13
2	115.00	11.00	25.05	71.00	74.25	20.00	1.06	5.00	0.05	19
3	116.78	9.00	24.14	52.00	64.75	24.00	0.98	5.25	0.05	17
4	109.28	8.00	24.04	60.00	67.63	22.00	0.99	3.75	0.05	13
5	113.53	7.00	26.51	57.00	60.38	30.00	1.01	3.75	0.05	17
6	115.78	8.00	26.30	57.00	69.00	33.00	1.21	4.25	0.05	21
7	115.78	9.00	25.85	54.00	97.00	24.00	1.47	8.25	0.05	17
8	115.53	9.00	24.91	63.00	87.50	16.00	1.18	4.38	0.05	20
9	107.28	9.00	23.94	60.00	86.63	18.00	1.20	6.75	0.05	17
10	111.53	10.00	24.30	62.00	64.63	22.00	0.95	3.50	0.05	17
11	115.53	9.00	26.37	68.00	82.63	19.00	1.16	5.13	0.05	20
12	106.00	7.00	23.84	59.00	70.63	27.00	1.12	3.88	0.05	10
13	113.28	8.00	23.85	50.00	57.63	29.00	0.94	3.00	0.05	10
14	107.53	7.00	24.57	75.00	87.13	24.00	1.32	8.00	0.05	18
15	116.28	10.00	27.42	71.00	87.88	21.00	1.27	8.50	0.05	18
16	106.28	9.00	24.29	74.00	88.63	27.00	1.41	4.75	0.05	22
17	116.78	9.00	25.01	62.00	62.50	27.00	0.99	4.75	0.05	24
18	114.53	10.00	23.26	56.00	99.38	27.00	1.58	9.25	0.05	21
19	104.53	10.00	23.95	60.00	84.00	19.00	1.18	4.25	0.05	25

Table 17 - Result distribution - Scenario 3 - Including movement groups